Freie Universität Berlin

Landmark-based Localization for Autonomous Vehicles

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von

Robert Spangenberg

Berlin
2015

Erstgutachter:	Prof. Dr. Raúl Rojas Freie Universität Berlin
Zweitgutachter:	Prof. Dr. Manfred Hild Beuth Hochschule für Technik Berlin
Tag der Disputation:	23. März 2016

Summary

Autonomous transportation will lead to major benefits in safety, economy and ecology. Although the associated technology has been an active field of research in the last decades, some problems have not been fully solved yet. Robust and efficient localization is a key component especially in urban scenarios. This thesis deals with the design and development of a system for landmark-based localization in urban scenarios suitable for autonomous driving. The sensor input is limited to a stereo camera pair, vehicle odometry and an off-the-shelf GPS. Prior knowledge in the form of a landmark map is also available.

Pole-like structures are identified as robust, long-term stable and common three-dimensional landmarks in urban scenarios. These are easily detectable by a stereo camera and are used as primary landmarks. In comparison to lane markers they have a lower occlusion probability and lower change rate. As pole-like structures can be rather small, a high quality depth reconstruction is crucial for robust detection. Several contributions are made in the field of automotive stereo vision, targeting long-term stability, robustness and efficiency. A new matching cost is presented and Semi-Global Matching is modified to become more reliable and more scalable. A robust extraction method for pole-like landmarks is introduced. The localization method proposed uses particle filters and the complete processing chain from feature extraction to processing a latency corrected vehicle pose output is covered. Field tests with an autonomous vehicle in urban environments and accuracy measures derived from real-driving data demonstrate the performance of the approach.

Zusammenfassung

Das autonome Transportwesen wird zu deutlichen Vorteilen im Bereich der Sicherheit, Ökonomie und Ökologie führen. Obwohl die dafür notwendige Technologie schon seit mehreren Dekaden ein lebhaftes Forschungsfeld ist, sind noch einige Probleme nicht vollständig gelöst. Die robuste und effiziente Lokalisierung ist eine Kernkomponente speziell in urbanen Anwendungsfällen. Diese Arbeit beschäftigt sich mit dem Entwurf und der Entwicklung eines Systems für landmarkenbasierte Lokalisierung im urbanen Umfeld, die für autonomes Fahren geeignet ist. Die genutzte Sensorik wird hierbei auf ein Stereokamerapaar, Fahrzeugodometrie und einen normalen GPS-Empfänger beschränkt. Vorwissen in Form einer Karte mit Landmarken ist ebenfalls verfügbar.

Pfahlartige Strukturen werden als robuste, langzeitstabile und verbreitete Landmarken in urbanen Räumen identifiziert und als primäre Landmarke verwendet, die leicht durch Stereokameras detektiert werden können. Im Vergleich zu Straßenmarkierungen haben sie eine geringere Verdeckungswahrscheinlichkeit und Änderungsrate. Da sie relativ schmal sein können, ist eine hochqualitative Tiefenrekonstruktion elementar für eine verläßliche Detektion. Mehrere Verbesserungen im Bereich der Stereoverarbeitung für die Anwendung in autonomen Fahrzeugen werden vorgestellt. Dabei werden Fragen der Langzeitstabilität, Robustheit und Effizienz addressiert. Ein neues Ähnlichkeitsmaß für dichtes Stereo Matching wird präsentiert und Semi-Global Matching modifiziert, um verlässlicher und skalierbarer zu werden. Eine robuste Detektion für pfahlartige Strukturen wird eingeführt. Die vorgeschlagene Lokalisierungsmethode benutzt Partikelfilter. Dabei wird die komplette Verarbeitskette, von der Landmarkenextraktion bis zur Berechnung einer latenzkorrigierten Ausgabe abgedeckt. Feldtests mit einem autonomen Fahrzeug in urbanen Umgebungen und auf echten Fahrsituationen beruhende Genauigkeitsmaße zeigen die Leistung des Ansatzes.

To my parents

It is far better to have absolutely no idea of where one is
– and to know it – than to believe confidently
that one is where one is not.

JEAN-DOMINIQUE CASSINI, ASTRONOMER, 1770

Acknowledgments

First and foremost I would like to thank my advisor Prof. Dr. Raúl Rojas. His vision of an autonomous car research project at the Freie Universität Berlin lead to the creation of AutoNOMOS Labs in 2006. I express my gratitude for giving me the opportunity to be a part of this research group. His constant support, guidance and pragmatism made the successful completion of this thesis possible. It was fascinating to take part in the development of the prototype car MadeInGermany, which has been driving autonomously on the streets of Berlin since 2011.

I would also like to express my gratitude towards the whole AutoNOMOS Team, that created two prototype cars and the whole ecosystem necessary to develop, test and operate autonomous cars.

I thank Prof. Dr. Raúl Rojas, Tobias Langner and Sven Adfeldt as co-authors of my publications for their very valuable insights. It was a pleasure to work with them. Thanks to Bennet Fischer and Lutz Freitag for our interesting discussions on vision and hardware topics. I am also indebted to Prof. Dr. Daniel Göhring and Fritz Ulbrich for numerous test or data collection drives and fruitful discussions on robotics, control or vision topics and their help with framework questions. I thank Daniel Seifert and Bennet Fischer for keeping up the common writing spirit, which was not always easy.

Susanne Schöttker-Söhl and Angelika Pasanec made sure I did not fall through the system as an external PhD student. I would also like to thank my employer Hella Aglaia for a very flexible part-time model, that allowed me to combine studies and work and the financial support to attend conferences.

My deepest gratitude to my mother, Prof. Dr. Marietta Spangenberg, for her optimism, pragmatism and constant will to provide feedback on the current stage of the project and its structure, especially during the process of writing this thesis.

For important hints at the final stage of this work I would like to say thanks to Dr. Michael Hähnel, my group leader at Hella Aglaia. His reflections have helped to increase the quality of this thesis. Most importantly, I want to thank my parents and my whole family for providing a great level of moral support in the rather long process of writing this thesis part-time and for always supporting my education and my work.

Contents

List of Symbols

$\alpha, \beta, \gamma, \sigma.$	algorithmic parameter
$\bar{E}(Q_l)$	average matching cost for disparity assignment l and induced valid pixels Q_l
$\dot{\psi}$	vehicle yaw rate
$\kappa(\cdot)$	Poisson intensity of a clutter process associated to a variable
$\mathbf{H}$	measurement matrix of a Kalman filter
$\mathbf{J}_f$	Jacobian of function f
$\mathbf{P}$	process noise covariance matrix of a Kalman filter
$\mathbf{Q}$	process noise matrix of a Kalman filter
$\mathbf{R}$	measurement noise matrix of a Kalman filter
$\mathbf{x}$	state
$\otimes$	bit-wise concatenation operator
ψ	vehicle heading
Σ	covariance matrix
a	axle distance of a vehicle
b	baseline of a stereo camera rig
$C(\mathbf{p}, d)$	data costs for disparity d at pixel $\mathbf{p}$
d	disparity

$d(\cdot), d(x, y)$	disparity map
$diag(d_1, \ldots, d_n)$	diagonal matrix of size n
E	UTM Easting
f	focal length of a camera
h	image height
$I(x, y)$	intensity image
$L_{\mathbf{r}}(\mathbf{p}, d)$	path cost for disparity d at pixel $\mathbf{p}$
N	UTM Northing
$p = (x, y)$	image point
$P = (X, Y, Z)$	world point
P_1	SGM penalty parameter for slanted surfaces
P_2	SGM penalty parameter for discontinuities
$S(\mathbf{p}, d)$	aggregated cost for disparity d at pixel $\mathbf{p}$
$s(u, v)$	a two argument sign function $s(u, v) = 0$, if $u \leq v$, $s(u, v) = 1$ otherwise
$s(x)$	a sign function $s(x) = 1$ for $x \geq 0$, $s(x) = 0$ otherwise
t	time
$T_{N_{eff}}$	threshold for number of effective particles of a particle filter
v	vehicle speed
$Val(l)$	fraction of valid pixels induced by assignment l
w	image width
$w_k^{(i)}$	weight of particle i at time step k

1 Introduction

The driverless car is an age-old human vision. This work is part of the general solution to the problem of localization of autonomous vehicles in urban environments. The goal is to analyze whether stereo vision with odometry and a low-cost GPS device can provide localization information at the level of quality needed for autonomous driving. In this process localization can use prior knowledge in the form of a map containing landmarks.

The maturity of the available solutions to the localization problem and for autonomous driving varies depending on the complexity of the environment. The available solutions are nearly ready to be used commercially on highways and in a few years time, on rural roads. In more complex scenarios like cities, it is still unknown when autonomous driving will become available. The concepts currently in place to deal with the full problem are several levels below human driving skills. Vehicles, even with the most advanced sensors, are not able to sense every detail of the unstructured, dynamic environments in urban scenarios. Based on their sensor readings alone, they are not able to navigate accurately and robustly enough in such complicated situations.

In addition to that, the usually programmed directive to follow traffic regulations completely and at all times cannot be applied in cities. Small violations of rules in order to keep up with the traffic flow are common and are also expected from autonomous vehicles. This means that an even more complete picture of the environment is needed, as the safeguard of regulations is loosened.

Therefore, the complexity of the problem needs to be reduced by decomposition. Autonomous vehicles need support with the use of prior knowledge in the form of maps. These maps can enrich the perception by adding reliable information layers that reduce the remaining complexity. They can contain the optimal trajectories at complex intersections, traffic light semantics or obstacles that are difficult to detect with on-board sensors. To be able to use these maps appropriately, localization is a crucial task.

Even current advanced driver assistance systems, like adaptive cruise control, lane keeping assist or city emergency braking can benefit from an accurate knowledge of the vehicle position. A better assessment of the structure of the vehicle's environment and the further course of the road is possible. This enables a higher level of accuracy and it

increases robustness of the assistance functions. Even if autonomous transportation is not yet readily available, ubiquitous localization can improve the intermediate solutions in the meantime.

Localization using laser scanner data is a solution often pursued. However, vision based sensors will also be necessary for autonomous driving, as the traffic environment is built for a mainly visual recipient, the human driver. A lot of traffic related pieces of information are transmitted only in the visual spectrum, like lane markers, traffic lights or signs. Previous approaches for the localization of autonomous vehicles with cameras concentrated on lane markers or general monocular feature detectors as input. As stereo cameras offer the possibility to gather visual as well as three-dimensional information they are valuable sensors in this scenario and more versatile than laser-scanners and monocular cameras.

The quality of the localization depends heavily on the properties of the landmarks used. The landmarks are to be detectable robustly with stereo cameras. Their position should be accurate enough to act as input for localization and they should be distinctive. A low number of false positives has to be attainable in urban environments and a high enough coverage, if only these landmarks are detected. Computational and storage efficiency are further key aspects. Landmarks should be easy to extract, match and store.

1.1 Thesis Statement

Localization for autonomous vehicles in urban environments is possible with a stereo camera, vehicle odometry information and an off-the-shelf GPS receiver using maps with landmarks. Pole-like structures can be detected reliably and efficiently with such a sensor setup and can serve as primary three-dimensional landmarks. The maps created are lightweight and compact. Every aspect can be demonstrated by experiments in real-driving situations in an urban environment.

1.2 Thesis Contributions

The contributions of this work are in the fields of localization and stereo vision for autonomous driving. We can summarize them as follows:

- For Stereo Matching:
  - Introduction of matching cost based online calibration, with a high-reuse of algorithmic parts from dense stereo matching.
  - A new matching cost is proposed, the Central-Symmetric Census Transform, leading to the same descriptiveness as state-of-the-art matching costs while providing more compact representation.

  - Weighted Semi-Global Matching modifies the way paths are integrated in the final aggregated costs, exploiting a prior on the three-dimensional structure of the modeled scene and leading to more robust matching results.
  - Rapid Semi-Global Matching improves matching speed by exploiting low- and high-level parallelism and adaptive disparity sub-sampling at negligible quality loss.

- For Landmark-based Localization using Stereo Vision:
  - A reliable extraction method for pole-like landmarks is presented, based on stereo vision disparity maps.
  - A localization method for autonomous vehicles using particle filters, including the complete processing chain from feature extraction to processing latency corrected vehicle pose output. Accuracy measures based on real data and field tests show that the system is able to perform at the accuracy level necessary and is robust enough for autonomous driving in urban environments.

1.3 Thesis Structure

The work is divided as follows. Chapter 2 covers the state of the art of stereo vision and gives an overview on related work in localization approaches for autonomous vehicles. Chapter 3 extends dense stereo matching approaches towards robustness, accuracy and real-time capability. Chapter 4 introduces a detection and mapping algorithm for pole-like structures from disparity maps. In Chapter 5, a Bayesian solution for localization is developed, based on the fusion of landmark maps, detected landmarks, odometry and GPS information. Experimental results for all topics covered are reported in Chapter 6, followed by the conclusion, including the list of key contributions of the thesis and directions for future work in Chapter 7.

2

State of the Art of Landmark Detection Based Localization

This chapter presents the state of the art related to landmark based localization for autonomous vehicles using cameras, with an emphasis on stereo vision based approaches. It starts with a condensed description of dense stereo matching and its usage for automotive scenarios. Quantitative evaluation results are presented and benchmarks in the desired application area are evaluated. The chapter then continues with landmark detection and mapping and closes with a survey of localization approaches for autonomous vehicles and a short commentary on the connected fields.

2.1 Dense Stereo Matching

Stereo vision aims to infer depth information from at least two cameras. The relative position of the cameras to each other is hereby stable and known. Their image acquisition is synchronized, leading to image pairs that are taken at exactly the same moment. Depth is calculated by means of triangulation, if corresponding points can be found in the two images. The goal of dense stereo matching is to calculate a dense depth map providing a disparity value for each pixel of the main camera input image. In contrast to that, feature-based stereo matching only computes depth values at sparse locations, but with higher reliability. Dense methods have to cope with informative and untextured image regions as well, where depth computation is difficult.

Correspondence search for a given image point is guided by the appearance of the matched point and its surrounding. The epipolar constraint allows the search area for each point to be reduced from 2D to 1D along the corresponding epipolar line. The standard form for stereo cameras aligns both cameras virtually to the same focal length and in a way that corresponding points are on the same scan line (Fig. 2.1). This process is called rectification and removes lens distortion as well. The stereo correspondence search outputs disparities in a map form, that are transferred to 3D points by triangulation.

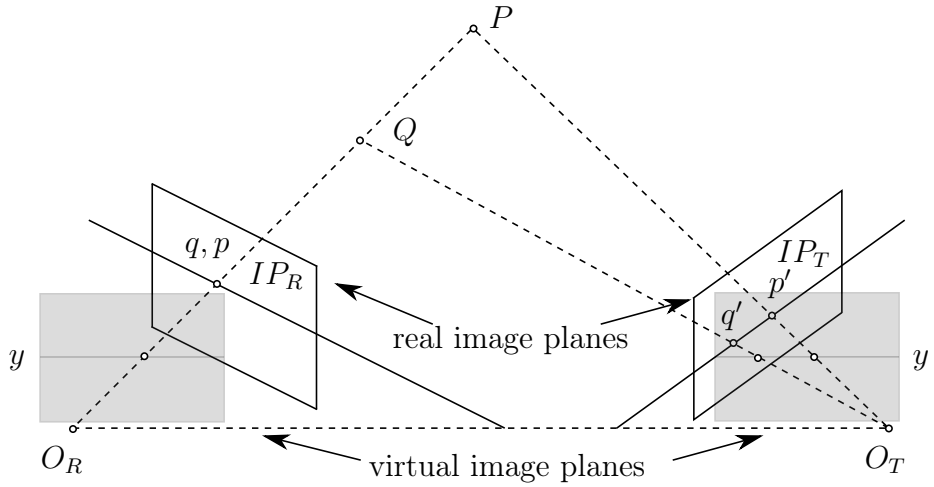


Figure 2.1: Stereo camera in standard form with aligned virtual image planes and the epipolar constraint

A prerequisite for rectification and triangulation is the knowledge of the cameras intrinsic and extrinsic parameters, which is done by calibration. Intrinsic parameters are the camera focal length, the image center and distortion coefficients. Extrinsic parameters describe the relative translation and rotation between the cameras. The resulting system has a data flow like in Fig. 2.3.

Triangulation works by using the baseline b, focal length f and the disparity d to calculate the world point (X, Y, Z) from the image point $p_R = (x_R, y_R)$ in the reference image (see Fig. 2.2):

$$Z = \frac{b \cdot f}{x_R - x_T} = \frac{b \cdot f}{d} \tag{2.1}$$

$$X = Z\frac{x_R}{f} = b\frac{x_R}{d} \tag{2.2}$$

$$Y = Z\frac{y_R}{f} = b\frac{y_R}{d} \tag{2.3}$$

Correspondence search in the standard form can be defined as the following problem: For a pair of rectified stereo images, with one image being the reference I_R and the other being the target image I_T, a corresponding point $p_T \in I_T$ needs to be assigned for each point $p_R \in I_R$. p_T lies on the same scan line as p_R and the search is constrained by a disparity range $D = [d_{min}; d_{max}]$. The process can be commonly divided into the following parts [97], using an approach that seeks to minimize costs for the found correspondences (see Fig. 2.8 for an example):

1. Matching cost computation

2. Cost aggregation

3. Disparity computation/optimization

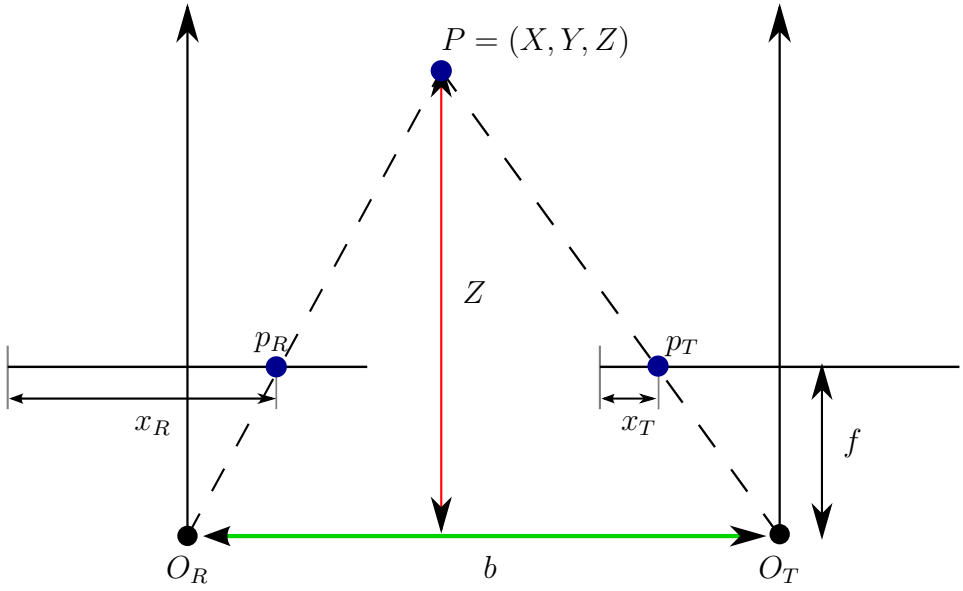


Figure 2.2: Stereo depth calculation by triangulation

4. Disparity refinement

Local algorithms perform the first three steps independently using a winner-takes-all strategy, whereas (semi-)global algorithms perform at least the cost aggregation step by minimizing a global cost function.

2.1.1 Stereo Matching Costs

A common and simple measure for the similarity of pixels is the absolute difference of their intensities (SAD):

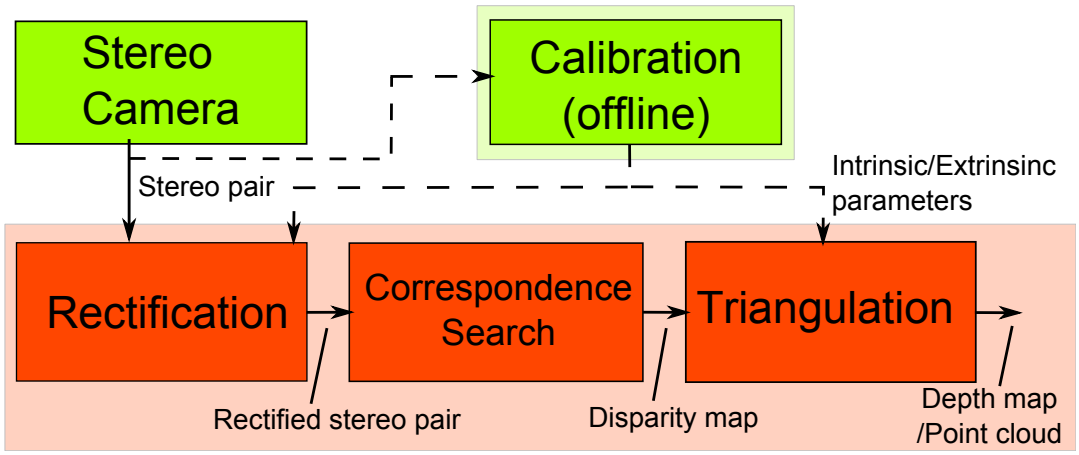
$$SAD(x, y, d) = \mid I_R(x, y) - I_T(x + d, y) \mid \tag{2.4}$$

or their squared differences (SSD):

$$SSD(x, y, d) = (I_R(x, y) - I_T(x + d, y))^2. \tag{2.5}$$

Those costs can be truncated to increase the robustness to outliers and can be aggregated over local support windows as well. Another measure is normalized cross-correlation (NCC), which correlates the two neighborhoods of the reference pixel and the test pixel with a compensation for mean and variance differences. This is prone to errors at discontinuities, as outliers lead to high costs in the calculation.

The non-parametric Rank transform [120] replaces the pixel with its rank among all pixels in its neighborhood. The idea is to increase the robustness to outliers in the neighborhood, occurring often at disparity discontinuities. Rank based costs (RC)



(a) Stereo system dataflow

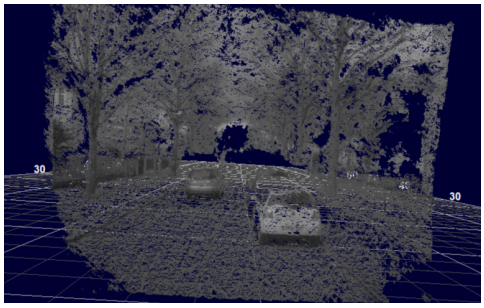
(b) Stereo pair left

(c) Stereo pair right

(d) Rectified stereo pair left

(e) Rectified stereo pair right

(f) Disparity map



(g) Point cloud

Figure 2.3: Stereo system standard dataflow with example intermediate results

simply measure the difference in rank:

$$RC(x, y, d) =\mid R_R(x, y) - R_T(x + d, y) \mid \tag{2.6}$$

The Census transform [120] is a non-parametric measure to describe each pixel using its intensity value in relation to its surrounding neighbors in a patch of $n \times m$ pixels. If the neighbor pixel has an equal or higher intensity value, the result is one, otherwise it is zero. With $s(u, v) = 0$, if $u \leq v$, $s(u, v) = 1$ otherwise, the calculation is as follows:

$$CT_{m,n}(x, y) = \bigotimes_{i=-n'}^{n'} \bigotimes_{j=-m'}^{m'} s(I(x, y), I(x + i, y + j)) \tag{2.7}$$

with $\otimes$ being a bit-wise concatenation and $n' = \lfloor n/2 \rfloor$, $m' = \lfloor m/2 \rfloor$. The Hamming distance of two Census transformed pixels is their matching cost. Usually, window dimensions of size 3×3, 5×5 or 9×7 are used, as their descriptions fit into 8, 32 and 64 bit.

2.1.2 Disparity Computation

Cost aggregation is necessary, as pixel-based costs are too ambiguous. Local algorithms aggregate over a local support window in order to reduce ambiguity. Thereby they assume a locally consistent disparity. The support windows can be of fixed size and form or adaptive. The result of the aggregation is the disparity space image (DSI), a three dimensional cost cube containing costs for all disparities for all pixels in the image plane. After aggregation, the disparity is selected by a simple winner-takes-all (WTA) rule:

$$d(x, y) = \underset{D_{min} \leq d \leq D_{max}}{\operatorname{argmin}} \operatorname{DSI}(x, y, d) \tag{2.8}$$

Global methods look for a disparity assignment that minimizes a cost function over the whole stereo pair, which imposes an additional regularization:

$$E(d) = E_{data}(d) + E_{smooth}(d) \tag{2.9}$$

The data term E_{data} is the appearance-based part, implemented by a matching cost. The smoothness term E_{smooth} is often modeled as a Markov Random Field and only the neighborhood of each pixel influences its smoothness cost. This penalizes disparity variations and only allows disparity changes at certain image features, e.g. intensity edges. As the minimization problem is NP-hard, approximations like Graph Cuts [18] and Belief Propagation [111, 62] have been pursued. Approximations over subsets of the stereo points are typically solved by dynamic programming along scan-lines, like Semi-Global-Matching (SGM) [45] or Simple Trees [15]. These semi-global methods result in a performance superior to the local ones without the drawback of the high computational requirements of the global approaches, which normally prohibit usage in real-time environments.

2.1.3 Disparity Refinement

The direct results of dense stereo matching algorithms contain outliers that should be invalidated or corrected to ease further processing steps. Furthermore, as the disparity resolution during matching is usually set to 1 pixel, a resolution increase is favorable. This is done through disparity refinement.

The reference frame during the correspondence search is not necessarily fixed. Either the left or the right image can be used as a reference and correspondences can be looked for in the other frame. Doing both searches results in two disparity maps d_{LR} and d_{RL}. It is now possible to validate them against each other by enforcing the following constraint on each of their pixels:

$$| \, d_{LR}(x, y) - d_{RL}(x + d_{LR}(x, y), y) \, | < T_d \tag{2.10}$$

T_d is often set to 1. This method is called Left-Right consistency check (LRC). Locations not passing this check are set to invalid. LRC reliably detects occlusions and preserves discontinuities [54], but increases the computational load significantly, as d_{RL} also has to be computed. However, an additional aggregation step can be saved by using a diagonal search in the aggregated cost volume for d_{LR} [124]. This leads to only slightly inferior results compared to full processing.

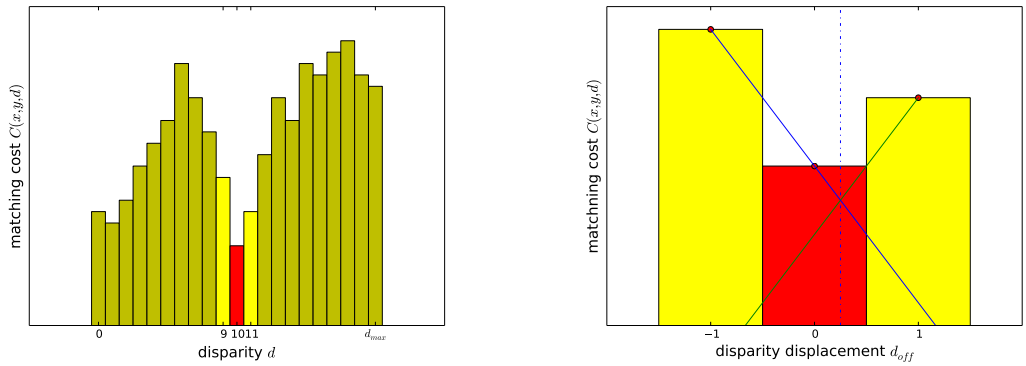
A lot of distinctiveness measures exist that attempt to evaluate the validity of a disparity by examination of the cost curve [54]. Those tests can be included in WTA. The Naive Peak Ratio (PKRN) test invalidates all disparities with an associated matching cost that is not significantly lower than the cost at the second minimum. The ratio of both costs is used to decide the distinctiveness. PKRN performs very well in the stereo case, especially with discontinuities [54].

Segmentation based approaches rely on the idea that the image can be segmented into regions with smoothly varying disparities. Speckle filtering segments the disparity map into connected components with smooth disparity changes. Regions with an area less than a specified threshold number of pixels are discarded as speckle and invalidated.

In order to facilitate further processing steps and in the case of applications requiring dense depth maps, invalidated disparities are interpolated from nearby locations. This can be simple background preserving methods using the minimum of the disparities of the surrounding valid pixels or discontinuity preserving methods [46]. Plane fitting approaches assume that all pixels in one segmented region lie on a plane [14]. This information is used to fill in missing disparities at invalidated locations. Segmentation can be based on the image intensities/texture and/or the disparity map.

Simple image filtering techniques are used as well, like median filters (e.g. with a 3×3 kernel), bilateral filtering [114] or approximations like the Guided Filter [41].

Typically, the sub-pixel disparity of a location is calculated by a parabolic or linear fit over the associated matching costs of the selected location and its neighbors (Fig. 2.4). It should be noted that the interpolation method has to match the matching cost, otherwise the so called pixel-locking effect occurs [102], leading to a bias towards discrete values.



(a) Cost volume slice for one pixel with minimum

(b) Sub-pixel offset d_{off} by equiangular interpolation at minimum disparity

Figure 2.4: Sub-pixel estimation by matching cost interpolation: The direct neighbours of the disparity with minimum cost offer information that can be used to compute sub-pixel disparities, here with equiangular interpolation.

2.1.4 Evaluation Benchmarks and Results

The Middlebury stereo data set [97] offers a methodology to compare different approaches using ground truth for studio scenes obtained by structured light. In connection with the proposed taxonomy, it was possible to compare different approaches on the same dataset. The influence cost functions was studied in [49] for three algorithmic classes: local correlation methods, SGM and a global method using graph cuts [18]. Rank-based costs are reported to be preferable for the local method. In connection with global radiometric differences and noise, mutual information is best for the semi-global and global methods. For local radiometric differences like vignetting, rank and Laplacian of Gaussian perform better than mutual information. SAD or SSD are inferior in all cases.

This analysis was extended in [50] comparing more cost functions, with the Census transform [120] as the best non-parametric cost and background subtraction using Bilateral filtering [3] as the best parametric cost. Census is superior to normalized cross-correlation in all cases. The usage of color information is tested as well and found to be not beneficial.

Usage of stereo matching for automotive applications like freespace evaluation and leading vehicle distance estimation was examined in [110] with SGM being better than local correlation and feature based stereo. The influence of sub-pixel calibration errors were examined in [47]. Census with a horizontal Sobel pre-filter to mitigate the de-calibration appears to be the most robust solution and is the best cost measure without additional adaption. Later, the KITTI benchmark suite [34] provided real-life driving scenes with Velodyne Lidar ground truth data and an online evaluation service. This covers a large disparity range ($d_{max} = 255$) and a wide variety of rural and complex inner city driving scenes. With uncontrolled lighting, reflecting surfaces, motion blur and poor contrast, this data set contains a lot of the challenges connected to autonomous driving.

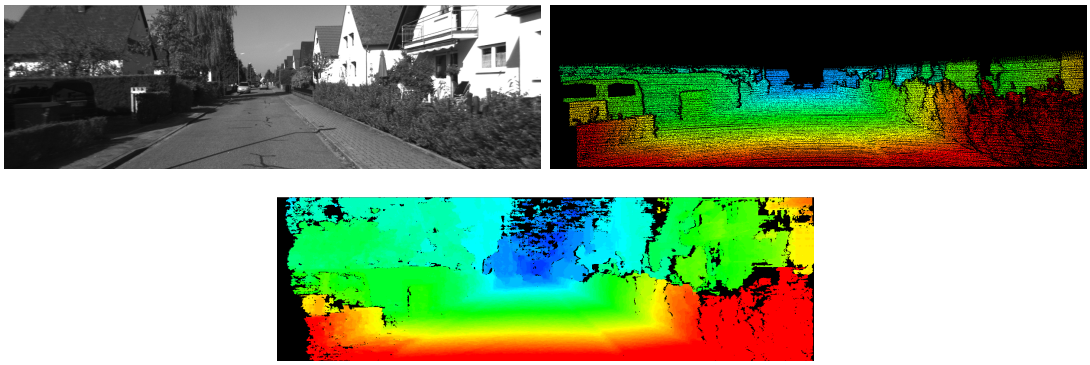


Figure 2.5: KITTI training data example: rectified left image, ground truth reference and baseline SGM results (top to bottom). Disparities are color-coded: warm - near, cold - far

This is divided into a testing and training data set with nearly 200 images each (Fig. 2.5). Rectified left and right images of a stereo pair are provided with ground truth based on accumulated lidar measurements. These are not completely dense and only available in the lower and middle region of the image. Ground truth is only accessible for the training part of the data set. The results of the evaluation for the part containing the testing data are determined by an online service disclosing the ground truth. SGM in connection with Census costs is one of the few real-time capable algorithms among the leading approaches in this benchmark (for an example result see Fig. 2.5).

The knowledge that it is used in automotive series production [32] and the evaluation results indicate that this method is a good starting point for research that aims to influence vehicle development. SGM provides a reliable, fast and high-quality base for feature extraction for localization. However, standard CPU implementations in high level languages are still demanding in terms of memory bandwidth and need several seconds processing time per frame.

2.1.5 Efficient Large Scale Stereo Matching

In efficient large-scale stereo matching [35] a search space reduction is achieved by a probabilistic generative model, which is based on support points. These points are chosen from image points on a rectangular sub-sampled grid, which are densely matched for stereo correspondence. If this matching is robust, points with a high uniqueness, sufficient local texture and with a consistent left/right-check qualify as support points. Furthermore, they have to share similar disparities with their neighboring support points as well to qualify as good representatives and not as outliers. The generative model forms a Delaunay triangulated mesh with robustly estimated disparities at the support points, which approximates the surface (Fig. 2.6). This model is now used to guide and limit the disparity search, leading to an efficient implementation and locally consistent results.

The quality of the depth maps is directly dependent on the ability of the algorithm to find robust support points for the generative model. If this step fails, large mismatched

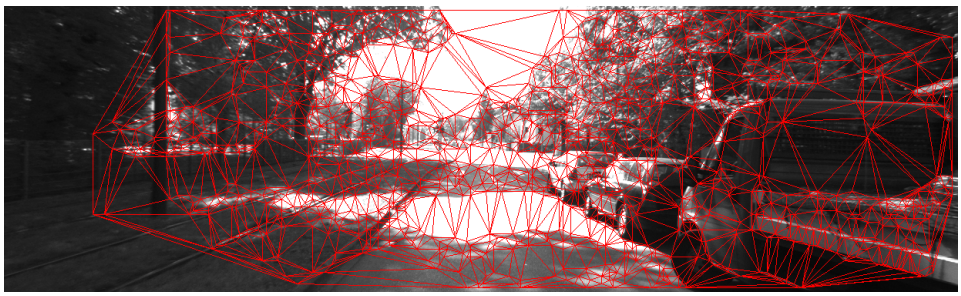


Figure 2.6: ELAS: Generative model described a Delaunay triangulated mesh (KITTI test set frame 112)

areas may appear. Furthermore, due to the sub-sampling during the creation of mesh, good feature points at the image borders might be missed. The triangulated mesh does not reach completely to the image rim and image borders. In particular corners have a high probability of not being matched at all.

2.1.6 Semi-Global Matching

The SGM algorithm by Hirschmüller [45] tries to approximate a global Markov Random Field regularized cost function by pursuing one dimensional paths L in various directions $\mathbf{r}$ through the image. Depending on the application, 8 or 16 paths are assumed to be necessary to cover enough structure of the image (Fig. 2.7a). This idea is an extension of previous dynamic-programming-based algorithms, that rely mainly on optimization over scan lines [13, 73, 16]. A common shortcoming of those approaches was a rather strong streaking effect, as the individual scan lines are not connected by the optimization. Each path has an associated minimum cost, which is determined by using dynamic programming:

$$\begin{aligned} L_{\mathbf{r}}(\mathbf{p}, d) = & C(\mathbf{p}, d) + \min(L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d - 1) + P_1 \\ & , L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d + 1) + P_1, \min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2) \end{aligned} \tag{2.11}$$

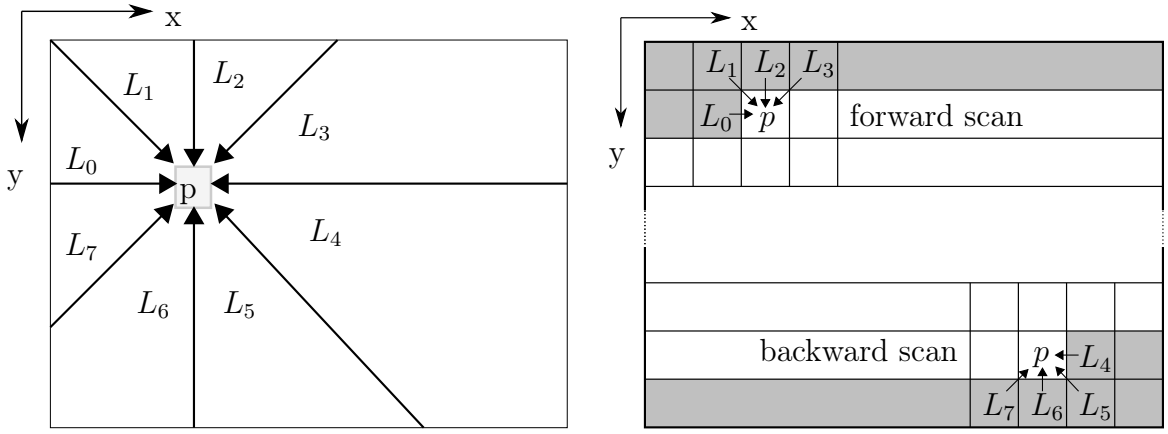
For every pixel $\mathbf{p}$ and disparity d, the cost is the sum of the matching cost $C(\mathbf{p}, d)$ and the cost of the minimum path to the previous pixel. The penalties P_1 and P_2 are regularizations. P_1 penalizes slightly slanted surfaces and P_2 depth discontinuities.

The costs from all paths are aggregated for all pixels and disparities resulting in the summed costs:

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d). \tag{2.12}$$

This summation connects the individual paths and reduces the streaking effect, as consistency information and appearance-based costs are propagated from several sides into each location.

The disparity value for each pixel is then selected by a winner-takes-all approach on S. Contrary to other solutions based on dynamic programming, explicit modeling of



(a) 8 paths (b) scans

Figure 2.7: Path directions in Semi-Global matching and computation implementation with two independent scans - grey areas are determined before the current pixel **p**

occlusions is not achievable. Therefore, a left-right consistency check is executed to invalidate wrong matches. The disparities of the right image d_{RL} are computed by the same approach or by using a diagonal search in S [45]. The data flow is depicted in Fig. 2.8.

The penalty P_2 can be adapted depending on the image content. Intensity edges in the image increase the probability of disparity jumps, thus the penalty should be reduced in such cases. The initially proposed inverse relationship between the penalty and the intensity gradient along the path [45]

$$P_{2,i} = \frac{\alpha}{\mid I(\mathbf{p}) - I(\mathbf{p}-\mathbf{r}) \mid + \beta} + \gamma \tag{2.13}$$

can be further simplified to a linear relationship as examined in [6]:

$$P_{2,l} = -\alpha \mid I(\mathbf{p}) - I(\mathbf{p}-\mathbf{r}) \mid + \gamma \tag{2.14}$$

without performance degradation. The adaption of P_2 should be limited to be bigger than P_1, otherwise slanted surfaces would be higher penalized than disparity discontinuities.

Implementations on Different Hardware Platforms

For SGM several real-time implementations exist for various computing platforms from classical CPU to GPU and FPGA solutions. The FPGA implementation in [22] makes use of low-cost to mid-range hardware. It achieves a frame rate of 33 Hz for a disparity resolution of 64 pixels and VGA sized images. Another FPGA approach processes slightly bigger images of 680×400 px at 25 Hz with a disparity range of 128 and a low

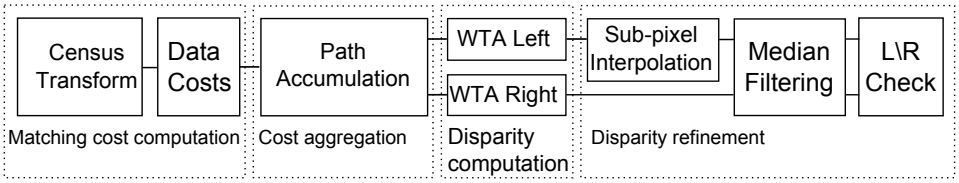


Figure 2.8: SGM dataflow

power consumption of 3W [32]. Memory bandwidth restrictions limit the system to two SGM engines that work on 64 disparities with 340×200 px image pairs. Both results are combined to a fused output.

GPU implementations process 11.7 fps for 64 disparities and VGA resolution employing a CUDA framework on a GeForce GTX 480 [75]. Another earlier solution reaches 13 fps on QVGA images, calculating a disparity range of 64 with a Cg based framework on a GeForce 8800 Ultra [29]. These GPUs require a lot of power and consume above 150 W.

CPU implementations aiming for speed are not readily available and publications are scarce. The proposal in [33] uses image and disparity sub-sampling to realize a processing speed of 14 Hz for an image resolution of 640×320 pixels. The disparity sub-sampling reduces the depth resolution in the close distance, but limits the depth uncertainty to stay below 1 m. SIMD (single instruction, multiple data) instructions compute 16 disparities at once at the path accumulation part, as accumulated costs are limited to 8 bit. This is only possible by using saturated arithmetic. Multiple cores are leveraged by OpenMP to evaluate each independent path in parallel. In order to evaluate the matching cost, a 5×5 Census transform is calculated, executed in parallel line-wise under the control of OpenMP.

2.1.7 Stereo Online Calibration

The notion of a rigid stereo rig, whose parts do not shift their orientations or positions over time, leads to a static camera calibration being used, obtained by methods using artificial reference patterns like checker-boards or circles [121]. In an automotive or machine vision use case, large temperature operation ranges, mechanical vibrations and material fatigue may cause long-term drifting of the camera parameters. Even changes to the dimensions of the housing, caused by small temperature fluctuations, may lead to significant errors in depth estimation, as pixel sizes are in the magnitude of one micron. This cannot be handled by thermal models alone. Only online self-calibration can adapt to this and make a long-term productive use of the stereo vision system free from any need of manual service possible. In addition, it may render the often costly and time consuming initial off-line calibration step unnecessary, enabling a simpler and more economical production.

A comprehensive method to online stereo-calibration for automotive vision is described in [26]. There, constraints emerging from recursive bundle adjustment are enforced. Furthermore, the trifocal constraint is used and pairs of stereo images allow the application

(a) FAST feature (b) SURF feature

Figure 2.9: Example detections of monocular features in urban scenes

of the epipolar constraint. The scale-invariant feature transform (SIFT) feature detector provides the features to impose the constraints on [67]. Fused by an Iterated Extended Kalman Filter they create an error tolerant framework even when using non-rigid stereo camera configurations. The approach of [65] relies on the epipolar constraint and applies the common basic steps in the self-calibration procedure. The output of a feature point detector is fed into a matching algorithm and matches are used for the computation of the associated Fundamental matrix, which contains information that can be used to recover the relative orientation. The structure tensor of the images is calculated and points with maximum "Minimum Eigenvalue" are selected input features. Positions of these features are not calculated with sub-pixel accuracy. Matching is modified by following a correlation based verification step, evaluating the correlation score to assess the uniqueness of the features.

2.2 Landmark Detection and Mapping

2.2.1 Monocular Features

Monocular image features have been quite popular for wide baseline estimation and image stitching, visual odometry or location estimation [115, 74]. The key concept is a scale and often rotation invariant feature detector coupled with a local descriptor. SIFT [67] was later simplified to Speeded-Up Robust Features (SURF) [9]. An early performance evaluation of local descriptors can be found in [76]. Maximally stable extremal regions (MSER) [70] perform a blob detection and extend the idea of connected components by intensity thresholding to a tree separated by maximal thresholds. Center Surround Extremas (CenSurE) [1] provide full scale resolution, improved speed and outperform SIFT and SURF in visual odometry. FAST [94, 95] was designed with computational efficiency in mind and produces stable features, but is not scale or rotation invariant (see Fig. 2.9 for an example of detected features).

Localization approaches for autonomous vehicles use a significant number of monocular features in order to achieve robustness [123] (in the region of a few hundred to a

thousand features per time slot). This comes with an increased computational load and bigger storage requirements for larger road networks. It is furthermore still unclear, how stable the features are over longer periods of time, while there has been some research on the influence of seasons have [115, 4].

2.2.2 3D Features

The majority of approaches are examined on very accurate point clouds, often devised by laser scanners or structured light. Spin images [58] provide a local feature descriptor for a 3D point cloud using oriented points. A "plane" rotated around the normal vector of the point collects images of all points belonging to the object in the vicinity of the feature point. The resulting 2D representation is rasterized and called spin image. Landmark detection using this descriptor was done on human models [10].

The rotation invariant feature transform (RIFT) provides a feature descriptor [104] and inspired by SIFT, uses local surface gradients instead of intensity gradients. Normal Aligned Radial Feature (NARF) [109] seeks to extract stable feature points that can be reliably detected from other perspectives. This is achieved by looking at dominant directions in surface orientation changes, non-maximum suppression and by limiting the points to be near strong distance changes. It furthermore provides a rotation invariant feature descriptor by building a star-shaped histogram.

Another possibility is the use of the local curvature of a surface. The orientation of a surface at a surface point can be described by the principal curvatures. Those are the maximum and minimum curvatures of the intersection of planes through the normal vector with the object surface. The Gaussian curvature $K = \kappa_1\kappa_2$, which is the product of the principal curvatures κ_1 and κ_2 or local curvedness measures [51] serve as feature selection criteria.

Stereo vision specific approaches are not so frequent. The stixel world representation describes the scene by rectangular super-pixels, called stixels [5, 86]. The aim is to model traffic scenes without relevant information loss and use this intermediate representation for subsequent classification and scene understanding. As only a few hundred stixels have to be examined in the later stages, computational load is drastically reduced. Detection of boundaries in stereo vision is done in [72], using occlusions in the resulting depth maps.

2.2.3 Lidar-based Features

Point feature histograms (PFH) [96] describe a point in a laser scanner point cloud by use of a histogram made up of its angular relation to its nearest neighbors. It has be successfully applied to point cloud registration.

The intensity response of a rotating lidar is used for correlation with a map in [66], providing reliable results even in dynamic situations. The detection of 3D objects, like planes and pole-like objects, is shown to be effective for vehicle localization in [21]. As these features tend to be sparser, the generation of combined super-features is proposed

Table 2.1: Vehicle localization approaches for autonomous vehicles: used features and testing areas

Features	Length, Environment	Ref.
Road Markings, Point Features	unclear, urban	[89]
Road Markings, Lidar, IMU, GPS,	8km, rural road	[98]
Radar, GPS, Road Markings	5km, rural road	[68]
Point Features, Road Markings, IMU	103km, rural/urban	[123]
Road Markings, GPS, IMU	3.5km, test track	[39]
Lidar, Road Markings, GPS	7km, rural road	[119]

in [99]. An advantage to the intensity-based solutions is the reduced dependency on weather conditions and an expected lower change rate of the features.

Direct correlation of occupancy grids provides very robust localization at the cost of higher computation times [83]. Hierarchical and GPU implementations make the approach real-time capable. A drawback is that the fraction of dynamic objects must not exceed a certain threshold, otherwise the estimation is too biased.

2.2.4 Landmark Mapping

While mapping of landmarks and driveable regions is often done implicitly as part of a localization solution [115, 123], an extended approach has been proposed in [11]. In addition to the description of geometrical components and a topological view regulatory rules are implemented, that describe the possible actions a vehicle is allowed to perform. Geometry is often reduced to two-dimensional approaches on the WGS84 ellipsoid used by GNSS.

Efficient location-based retrieval of landmarks, road surface sections that can be driven on and points of interest are often realized using space-partitioning trees. k-d-Trees are a popular option. Team AnnieWAY used them during the 2011 Grand Cooperative Driving Challenge [36]. Oct-Trees [52] or R-trees [11] are leveraged as well. Landmarks can be seen as an additional object layer stored in these maps.

2.3 Localization for Autonomous Vehicles

Approaches using an offline map often rely on Bayes filters to integrate odometry data with external information sources as lidar, radar or vision features and GPS (overview in table 2.1). Multiple approaches exist and only a selection can be summarized here, these are examples of localizations in dynamic, uncontrolled situations on public roads.

The approach in [89] uses road markings extracted from aerial photographs to create a feature map, which is used for localization in urban scenarios using an onboard stereo camera rig. This also extracts road marking features. Short term unavailabilities due to non-marked areas are mitigated by visual odometry. A sigma-point filter implements a recursive state estimation. Evaluation is done relative to a INS/GPS system at a city intersection. The longitudinal error has a mean of $-0.895m$ and standard deviation of $2.26m$, while the lateral error is significantly more accurate with $0.097m$ mean error and a standard deviation of $0.815m$. The mean orientation error is $-2.3395°$ with a standard deviation of $1.1588°$.

A combination of laser scanner information with lane markings, GPS and vehicle dynamics is fused by a particle filter in [98]. Lane marking measurements and laser-scanner landmarks are brought together with information in a high-precision digital map. The road marking association is done by prototype fitting. On the rural test track of 4 km length, driven in both directions, the results are compared to DGPS-RTK data. The localization error stays below 1 m and the orientation error below 1° at all times and the system is real-time capable. The lateral position has a mean offset of 0.12 m and a standard deviation of 0.096 m. The heading is 0.266° off on average with a standard deviation of 0.185°.

The approach in [68] uses an extended Kalman filter to integrate features provided by radar targets, lane marking information and GPS, provided by off-the-shelf sensors. They state that all information sources are necessary to stay below a lateral threshold of 20 cm in the majority of the situations on the test route. The reported accuracies are measured relative to a DGPS reference. In 87% of the situations, the system has a lateral position error lower than 0.2 m and in 94% a longitudinal position error lower than 1 m. Both thresholds are stated as goals to allow autonomous driving.

The sigma-point filter in [123] integrates visual features and structural road information as curbs and markings with data from an inertial measurement unit (IMU). Visual features were detected using a front and a rear camera. The lateral accuracy is not measured directly, but had to be in the region of 10 cm to 20 cm to master the test route. As a few hundred to a thousand features per frame are used for feature based localization, the approach is computationally very intense. Storage requirements are high, as all matched features with position and descriptor have to be stored. How many of those features are needed for reliable long-term localization and how stable they are is a topic of ongoing research.

In [39] lane markings provide mainly lateral and orientation information. Two downward-facing side cameras mounted on the roof detect the markings. This information is fused by an Extended Kalman Filter to GPS and IMU data, reaching a mean lateral offset below 10 cm in the experiment, which was conducted on a test track.

A particle filter is used in [119] to fuse lidar-based pole-like objects with lane marking parts detected by one front and one rear-facing camera and odometry information. The lane marking parts are extracted by a Maximally Stable Extremal Region (MSER) detector [70] directly from the camera image. The lidar data is first processed to an occupancy grid, which is searched for MSER features as well. The accuracy of the approach is measured relative to a DGPS-RTK system and calculated by the analysis

of 50 different runs of the particle filter. The lateral error on the rural test track has a mean of $0.08m$ and a standard deviation of $0.11m$. The longitudinal error has a mean of $-0.06m$ and a standard deviation of $0.16m$. The computational load is moderate, as it is reported that the 15 Hz update rate of the lidar can be processed at all times. The storage requirements are very low, as only the positions of a dozen features have to be saved and processing load is moderate.

Reported accuracies can be seen as an indicator for system performance. However, the often used differential GPS with real-time kinematic (RTK) reference systems deliver absolute accuracies with a standard deviation of 0.15 m for position, with outliers in the range of 1m, e.g. due to limited satellite availability in urban scenarios. Even the usage of own reference stations does not improve the situation very much, as a distance to the reference station of 2500 m may lead to position differences of at least 0.05 m ([117] p. 46). The influence of atmospheric differences that disturb the signal of the two GPS receivers in the reference system becomes a major source of errors even at those limited distances. The accuracy of the measurement system might not be sufficient to reliably estimate the accuracy of the localization system. As DGPS-RTK systems are among the best reference systems available for position estimation of moving platforms, their limitations have to be accepted. Other approaches, like marker-based localization or external camera-based solutions are usually not feasible on a larger scale.

2.4 Connected Fields

Simultaneous Localization and Mapping

SLAM has been a very hot topic of research in the robotics community for several decades. The aim is to localize a robot and build a map of the localization information seen, usually landmarks, at the same time. As in most cases it is only local position information that is available, error bounding relies on successful loop closures. The robot has to detect whether it has entered an area that it has visited before. In this case, the odometry uncertainty between both visits can be reduced significantly by exploiting the loop closure. The research field started with [105], stating the problem and offering the first solution in the form of an extended Kalman filter.

Later, real-time oriented solutions based on particle filters such as FastSLAM [78] or FastSLAM 2.0 [77], provided robust and simple solutions to false associations. These algorithms are covered in depth by Thrun in [113]. With the incorporation of GPS information and additional insights into the problem structure, it was possible to modify the SLAM solutions to enable the mapping of large-scale urban structures [23]. The classical sensor for SLAM is a lidar range scanner. Stereo rigs are used a lot less, e.g. in [74]. Here a real-time solution is implemented with the drawback of omitting the search for loop-closures. A relative SLAM approach is used. This solution does not pursue globally bounded errors, but instead seeks to minimize local errors. One trend is a shift towards relative topometric methods as [103] proposes, which even support unobservable ego-motion.

SLAM has made extreme progress in the last years, but the problem is far from being completely solved. Real-time capability, robustness and use in dynamic, urban environments together with a vision-based sensor, as is our use case, still seem currently out of reach. The use of a DGPS with RTK data for mapping and subsequent localization using this map is the more realistic research goal.

Visual Odometry

Visual odometry deals with the estimation of the movement of a platform equipped with a camera or a camera rig based on video input alone. No further information sources as wheel encoders or GPS are used. Typical setups include monocular cameras [81] or rely on stereo rigs [63, 53, 88] to increase robustness. Full independence of translation and rotation constraints and a minimized drift is a research goal, in connection with the fulfillment of real-time constraints.

A frequent first step is the detection of feature points in each frame of the video input. The found features are matched with subsequent frames based on their descriptors and tracked, if possible. In non-static environments, multiple solutions might occur. In order to mitigate their influence, RANSAC schemes [81] or inlier detections based on stereo constraints are employed [48, 53]. A robust estimation of the movement of the camera based on the tracked features follows. As the relative motion between two frames might be too small, integrating the information over more frames is advantageous. The use of online bundle adjustment [64] to increase robustness is an approach which also seems favorable. All cited solutions exhibit real-time capabilities.

Despite their short term accuracy, current solutions suffer from long term drift, starting at drives of more than a few hundred meters. Without other reliable information sources for absolute positioning, such as GPS, only locally smooth results can be obtained. A correction of small angular errors, e.g. in tight bends, is in principle not possible, this leads to large absolute positional errors without a global reference. As the goal is only relative localization, no map is produced. Only if the complete trajectory were kept for bundle-adjustment, could this influence be bounded and a map could be created. Nevertheless, a method based on visual odometry could be used as the starting point of an incremental mapping process or as a smoothing part in combination with GPS, delivering the relative positions for the elements of the map.

3

Automotive Stereo Vision

This chapter details the parts of the chosen approach to quickly get a reliable and robust depth map in order to be able to detect landmarks for localization and mapping. As, over time, the stereo rig is not as stable as necessary, an online calibration method is proposed, that re-uses a lot of parts of the dense matching algorithm and is based on the optimization of matching costs. Based on the state of the art analysis, Semi-Global Matching in connection with Census-based matching costs is used a procedure for stereo matching. This approach is improved in several ways. A variation of the Census-based matching cost is designed using ideas from Local Binary Patterns. Hereafter, the aggregation step of Semi-Global Matching is modified with the goal to increase robustness by reducing the influence of wrongly propagated disparities. The chapter closes with modifications targeting processing load reduction and efficient use of multiple core and multi-instruction architectures. The influences of using the algorithm on image stripes are mitigated and a different view of the algorithm enables a low-overhead sub-sampling strategy in the disparity dimension.

3.1 Stereo Online Calibration

Both approaches to stereo online calibration reviewed in the state of the art chapter (see subsection 2.1.7) treat the calculation of the disparity image by stereo matching and the self-calibration method as separate tasks. No or only little reuse of the algorithmic parts of stereo matching is possible and the remaining extra effort for implementation is sizable.

In contrast to the normal approaches, the accuracy of the calibration is measured by a cost based on the outcome of the dense stereo matching (approach first presented by myself in [106]). At first I look into measures that are easily available as by-products or results of the matching algorithm - matching costs and the fraction of pixels not matched. Secondly, I propose a scheme to efficiently lead the search process through calibration space using a Markov-chain Monte Carlo approach.

3.1.1 Calibration Parameters of Interest

It is presumed that the internal calibration of both cameras in the rig is stable and known. Additionally, an initial assumption on the values of the external calibration is available. The relative rotation and translation of the stereo cameras is of interest. A comprehensive sensitivity analysis in [26] and my own practical observations show strong correlations between small changes of principal point positions and extrinsic orientation parameters. The length of the base of the stereo rig cannot be estimated by image-based observations only. Thus, for a static camera set-up, the computation of offsets for the three relative angles yaw, pitch and roll is a realistic and sufficient goal.

3.1.2 Measuring Calibration Accuracy

No knowledge of the observed scene is assumed, therefore no ground truth is available. As stereo matching is typically modeled as an energy minimization problem, I am able to use the connected costs as an evaluation measure.

The process of dense stereo matching can be tackled as a labeling problem. Let D be a set of finite size that contains the disparities and let P form the set of all image pixels. Then l is the outcome of a dense stereo matching and relates every pixel $p \in P$ to a disparity $l_p \in D$. The excellence of this labeling can be assessed by the energy function $E(l, P)$:

$$E(l, P) = \sum_{p \in P} \left(D_p(l_p) + \sum_{q \in N(p)} W(l_p, l_q) \right) \tag{3.1}$$

hereby forms $N(p)$ the 4-way neighborhood of p, $D_p(l_p)$ describes the data costs of the assignment of l_p to p and $W(l_p, l_q)$ is a cost measure between l_p and its neighbor pixels l_q representing the smoothness costs. Typical stereo matching algorithms try to calculate this cost either in a local or global optimal way. Simple algorithms neglect the $W(l_p, l_q)$ part and only depend on the data cost part. If Q_l is the set of pixels that have a valid disparity in the assignment f, then

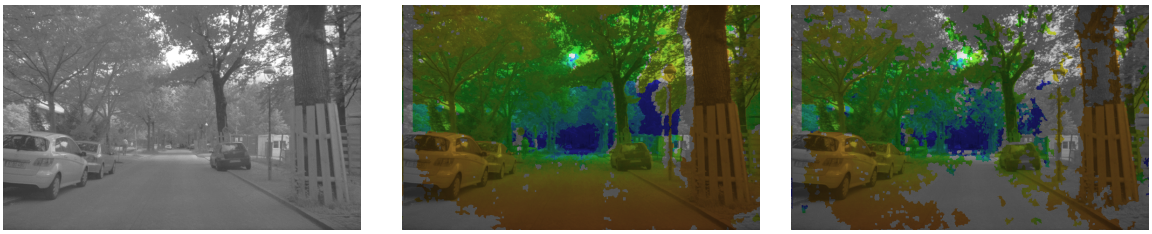
$$\bar{E}(Q_l) = \frac{E(l, Q_l)}{|Q_l|} \tag{3.2}$$

is the mean matching cost per valid pixel. This measure should increase, if the calibration is changed in a way that is less correct.

Calculating the cost of the selected assignment is quite straightforward for the majority of the algorithms. For some of them, the matching cost might not be easily available or may lead to a performance degradation, e.g. in a FPGA or GPU solution. In these situations, the percentage of pixels with a valid disparity could serve as a measure for calibration quality

$$Val(f) = \frac{|Q_l|}{|P|}. \tag{3.3}$$

Again $Val(l)$ should be higher in comparison to $Val(l')$ of the assignment l' with the

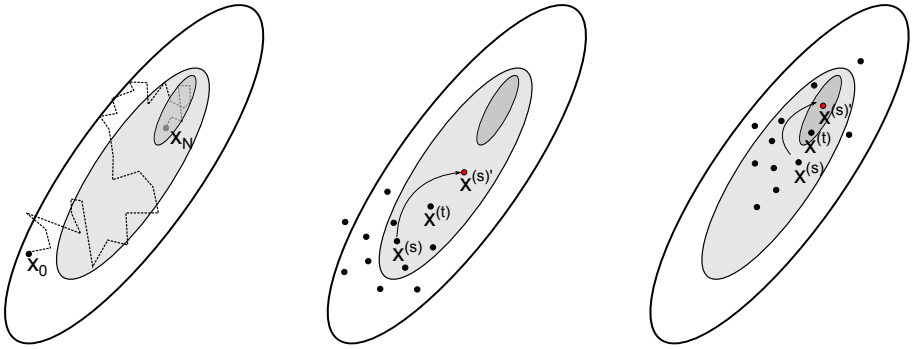


(a) Campus scene

(b) Depth map with optimal calibration

(c) Depth map with pitch offset of -0.1° and roll offset of 0.1°

Figure 3.1: Influence of miscalibration on depth map and the percentage of valid pixels.



(a) Metropolis walk

(b) Skilling: Initial step

(c) Skilling: After a few steps

Figure 3.2: Markov-Chain Monte-Carlo: Metropolis walk vs. Skilling's method. It is obvious that in this elongated density, the Metropolis walk is quite efficient, while the other method adapts to the density in the course of the iterations.

correct calibration (Fig. 3.1).

3.1.3 Search Procedure

In order to robustly find the optimum of the calibration accuracy functions, Markov-chain Monte Carlo approaches are a viable alternative compared to gradient descent approaches. MacKay [69] presents the Leapfrog-method by Skilling as an extended version of the standard Metropolis algorithm. The Metropolis algorithm simulates a Markov-chain with the probability density $P(\mathbf{x})$. This is done using a symmetric proposal or jumping density $Q(\mathbf{x}'|\mathbf{x}_t)$, which is in comparison simple to compute. The result of the algorithm is a set of samples S=$\{\mathbf{x}_t\}_{t=0\ldots N}$.

Metropolis-Algorithm

1. The first sample $\mathbf{x}_0$ is chosen randomly

2. Given $\mathbf{x}_t$ one computes a $\mathbf{x}_{t+1}$ by:
   - $\mathbf{x}'$ is sampled from the density $Q(\mathbf{x}'|\mathbf{x}_t)$.
   - Calculate acceptance ratio $\alpha = \frac{P(\mathbf{x}')}{P(\mathbf{x}_t)}$.
   - $r \in [0, 1]$ is a uniform random number
   - $\mathbf{x}_{t+1} = \begin{cases} \mathbf{x}', & \text{if } \alpha \geq r \\ \mathbf{x}_t, & \text{else} \end{cases}$

The expectation is that the Markov-chain, simulated by the algorithm, has a high probability to visit regions near maxima in state space. Especially for elongated densities, the Metropolis algorithm is slow in the exploration of state space and therefore inefficient. Thus, I employ the Leapfrog-method by Skilling. It is a variation of the Metropolis algorithm using a small number of state vectors $\mathbf{x}^{(s)}$ simultaneously instead of only one state vector $\mathbf{x}$. A state vector $\mathbf{x}^{(s)}$ is modified to $\mathbf{x}^{(s)'}$ with a partner state vector $\mathbf{x}^{(t)}$ by:

$$\mathbf{x}'^{(s)} = \mathbf{x}^{(t)} + (\mathbf{x}^{(t)} - \mathbf{x}^{(s)}) = 2\mathbf{x}^{(t)} - \mathbf{x}^{(s)} \tag{3.4}$$

The partner state $\mathbf{x}^{(t)}$ is chosen either at random or using a weighted distance function. The detailed balance condition has to be fulfilled if weighted distances are used, in order to not distort the simulated density. This method improves the sampling by adapting better to the local shape of the approximated density, as Fig. 3.2 shows. This leads to more efficient walks through state-space and no proposal density has to be specified, which is sometimes difficult.

The simulated chain S of the Leapfrog-method is evaluated in order to gain the cost minimum. A local deterministic search near this point in state space is performed to gain additional accuracy.

The experimental evaluation and field test results for the stereo online calibration are located in subsection 6.2.1 in the evaluation chapter.

3.2 Center-Symmetric Census Transform

In order to reduce the computational load during the calculation of the Census Transform, a sparse version has been proposed [124], that simply sub-samples the image patch. Fewer bits in the descriptor mean fewer bits as inputs for the Hamming distance as well, which speeds up the matching cost calculation and can enable the usage of specialized hardware instructions as POPCNTs.

The Local Binary Pattern (LBP) operator [82] transforms each pixel utilizing the relative gray intensities to its surrounding. If the neighbor pixel intensity is equal or higher, the transform result is one, otherwise it is zero. The pixel result is the concatenation of the results for all neighbors. They are grouped in a binary string coded as a single number. The calculation can be defined with the sign function $s(x)$ as follows ($s(x) = 1$

for $x \geq 0$, $s(x) = 0$ in all other cases):

$$LBP_{R,N}(x,y) = \sum_{i=0}^{N-1} s(n_i - n_c)2^i \tag{3.5}$$

where n_j is the intensity of a pixel j of N equally distant pixels on a circle of radius R with center (x, y) and n_c is the intensity of the center pixel. Bilinear interpolation is used to calculate the intensities of neighbors that are not exactly located on a pixel.

Center-Symmetric LBPs [42] use a more compact description, storing only comparison results of center-symmetric pairs of pixels. Furthermore, a comparison threshold T is added:

$$CS\text{-}LBP_{R,N,T}(x,y) = \sum_{i=0}^{(N/2)-1} s(n_i - n_{i+N/2} - T)2^i \tag{3.6}$$

Using the idea of Center-Symmetric LBPs to compare opposite pixels on the region, as first presented in [107], I now propose the Center-Symmetric Census Transform (CS-CT) as

$$CS\text{-}CT_{m,n}(x,y) = \bigotimes_{(i,j)\in L} s(I(x-i, y-j), I(x+i, y+j)) \tag{3.7}$$

with L being defined as follows. Let $n' = \lfloor n/2 \rfloor$ and $m' = \lfloor m/2 \rfloor$ be the floored halves of the patch width and height. $R_{a,b}$ is an interval of integers with boundaries a and b:

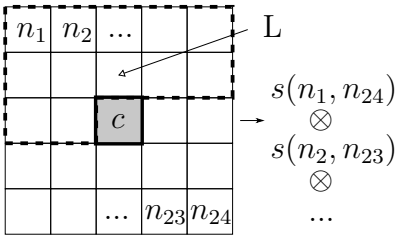
$$R_{a,b} = \{x \in \mathbb{Z} | a \leq x \leq b\} \tag{3.8}$$

L is the union of the two tuple sets L_1 and L_2: $L = L_1 \cup L_2$. L_1 covers tuples in the left upper corner $L_1 = R_{-n',0} \times R_{-m',0} \setminus \{(0,0)\}$ and L_2 is located in the right upper corner $L_2 = R_{1,n'} \times R_{-m',1}$. Due to symmetry, no more tuples are necessary.

The shared idea with CS-LBPs is that only center-symmetric pairs of pixels are used in the calculation, but in a Census-like way an image patch of $n \times m$ is used (Fig. 3.3a). As with the Sparse Census transform, CS-CT has a representation of 31 bits which describe a patch of 9×7 pixels, but covers more pixels within the calculation. As the comparisons are not dependent on the center-pixel, and the compared pixels change, the noise dependency of the measure should improve as well.

The saved bits may be used to implement a weighted Hamming Distance (Fig. 3.3b) through bit duplication. This integrates easily with implementations utilizing hardware bit count instructions as population counts for the Hamming Distance. Otherwise, weighting can be realized without additional bits by using modified lookup tables. The idea behind the introduction of weights is to reduce smearing effects that appear especially at the borders between regions with less textured and highly-textured regions.

The experimental evaluations are located in subsection 6.2.2 in the evaluation chapter.



(a) Construction sketch illustrating L

1	1	2	1	1
1	1	2	1	1
2	2	c	2	2
1	1	2	1	1
1	1	2	1	1

(b) Additional pixel weight structure along central rows and columns

Figure 3.3: Center-Symmetric Census Transform for a 5x5 patch

3.3 Weighted Semi-Global Matching

Semi-Global Matching by Hirschmüller strives to approximate a global MRF regularized cost function by pursuing one dimensional paths L in multiple directions $\mathbf{r}$ through the image (see section 2.1.6).

The cost information from all paths is integrated by summation of all pixels and disparities resulting in the accumulated costs

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d). \tag{3.9}$$

This sum is motivated by the assumption that all paths propagate the same information content to the aggregated costs. However, some paths may contain less or even wrong information, as they might pass depth discontinuities or less textured areas. This can lead to the false propagation of disparities into untextured areas. A simple sum of all paths will only be correct if the fronto-parallel assumption holds for the surrounding area of the current pixel.

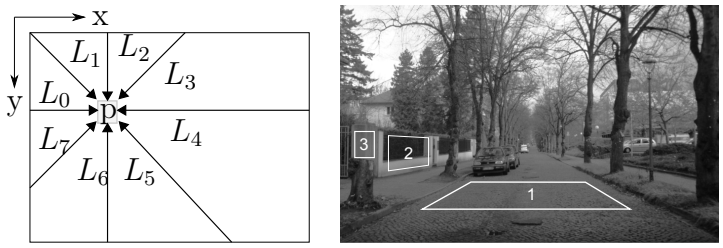
Furthermore, if the assumption of the path, that the disparity is constant or only slightly changing is fulfilled, it will cover the local structure a lot better than the other paths for that location.

I propose a method named Weighted Semi-Global matching (wSGM, first presented by myself in [106]), which weights the cost of each path depending on its compliance with the related surface normal

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} W(\mathbf{r}, \mathbf{p}) L_{\mathbf{r}}(\mathbf{p}, d). \tag{3.10}$$

The rest of SGM remains unchanged.

A plane P describing a three dimensional surface patch is assumed. The vanishing line of P under central projection will concur with the vector pointing along the direction of constant disparity values on this plane. Hence, the algorithm should propagate disparities preferably along paths neighboring this direction. This is realized by raising the weight of SGM paths depending on the angle between the path and the local vanishing



(a) 8 paths (b) Different types of surfaces

Figure 3.4: The path directions in Semi-Global Matching propagate disparities. The optimal choice of surface dependent weights depends on the orientation of the surface.

line. This increases the weight of the path in the aggregated cost.

In a road scene (Fig. 3.4b) the locations on the road surface (area 1) should exhibit a nearly constant disparity along the horizontal paths L_0 and L_4. Thus, raising of the weight for these paths is beneficial for consistency. Structures parallel to the road that are vertical should benefit from raised weights for the vertical paths (area 2). In contrast to this, frontal-parallel structures should use the same weights for all paths (area 3). However, in many cases the surface normals are not known and as one tries to recover the surface by matching, a chicken-egg situation is encountered.

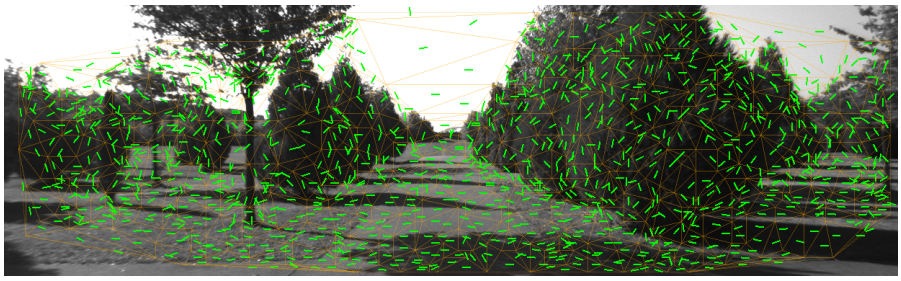
Two different approaches were pursued to solve this. The first one applies normal SGM on a sub-sampled image to compute an initial surface estimate. This can be used to calculate the weights for the full image. The second approach is derived from a part of ELAS (see section 2.1.5). The generative model is used to estimate the surface normals (Fig. 3.5) and then choose the weights:

1. Match on sub-sampled grid
2. Select robust support points
3. Perform Delaunay triangulation to create mesh
4. Calculate surface normals and preferred direction per triangle
5. Classify triangles to choose weights

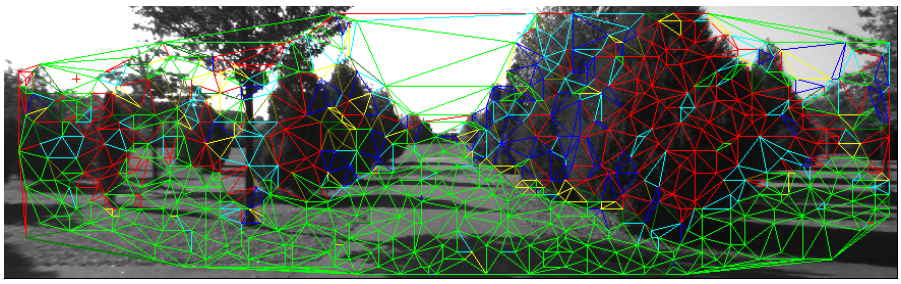
For all image points the weight modification can be carried out, if they are inside the mesh. The remaining relative weights are chosen to be equal. As the process selects a robust subset of points for the normal calculation, the probability of outliers should be low for the surface normals. The additional cost of the preprocessing step is in the region of 10% of the load of the normal matching.

The improvements can be seen in Fig. 3.6. The algorithm is now able to reconstruct the weakly structured road surface without the erroneous step caused by the normal aggregation. Further experimental evaluations are located in subsection 6.2.2 in the evaluation chapter.

(a) KITTI training frame 9 with grid used for sub-sampling



(b) Triangulation and preferred directions



(c) Assigned directions

Figure 3.5: wSGM preprocessing steps to infer path weights: Of all points in the grid, reliable support points are selected and triangulated. The local plane normals are calculated and finally the preferred directions are selected.

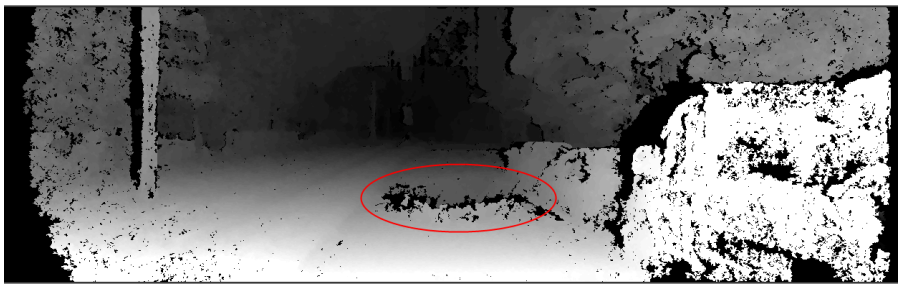
3.4 Rapid Semi-Global Matching

The idea of Rapid Semi-Global Matching (rSGM, first presented by myself in [108]) is to develop a high speed modification of SGM for implementation on a multi-core CPU for prototyping driver assistance applications. The parts of the algorithm shall be adapted to the parallelization possibilities of the platform with minimal quality degradation and with the goal of real-time execution at VGA image resolution with a large disparity range.

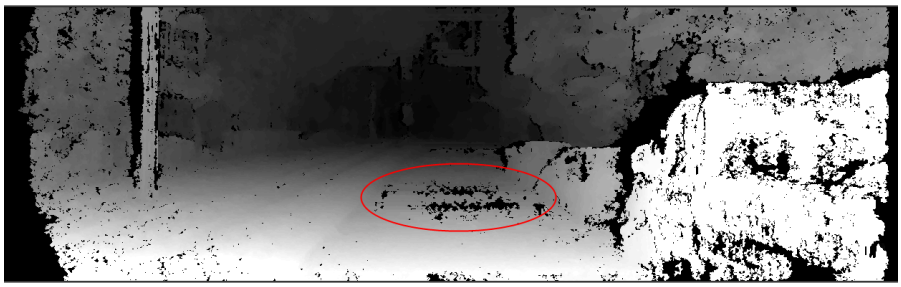
3.4.1 Algorithmic Structure and Parallelization Concept

The SGM algorithm parts are decomposed as follows. At first the matching costs are computed; Census costs, a 5×5 or 9×7 Census window with a Hamming Distance (em-

(a) Low texture road surface



(b) SGM disparity map



(c) wSGM disparity map

Figure 3.6: KITTI test data set frame 112: false disparities are propagated by SGM and invalidated with wSGM as can be clearly seen in the marked textureless area.

ployed by [55] as well). Then, path accumulation is executed with 8 paths. In automotive applications, the variant using 16 or more paths only leads to minor improvements. This does not justify the greater computational effort. The path costs are aggregated to form the disparity cost cube. Using this cube, the disparities leading to the minimum costs are chosen for both viewpoints. This is achieved by a simple but effective winner-takes-all (WTA) strategy with minimum evaluation. Both images are filtered using a median filter and occluded regions and mismatches are invalidated by the left-right-consistency check. To sum up, the following computational parts are used:

- Census mask computation
- Data cost calculation
- Path accumulation

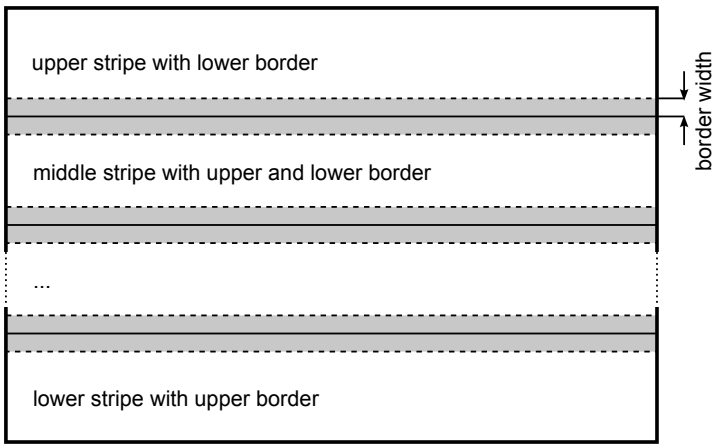


Figure 3.7: Image stripes overlap through an additional border, in order to reduce the information loss caused by striping

- WTA left and WTA right
- Sub-pixel interpolation
- Median filtering
- Left-right-consistency check.

Normal CPUs offer basically two options for parallel processing: data-level parallelism by single instruction multiple data (SIMD) instructions ([43],chapter 4: p. 264ff.) and thread/core-level parallelism ([43], chapter 5: p.344ff.) as they usually offer multiple cores that share the same main memory . SIMD instructions can be leveraged at the following parts: Census transform, Data cost calculation, Path accumulation, Winner-Takes-All and Median Filtering.

The question arises as to how to use the thread-level parallelism. Apart from the path accumulation step, image lines can be calculated independently parallel without any additional synchronization cost and increased memory usage. Horizontal image stripes form blocks of lines and assigning one stripe to each thread seems a favorable strategy. Path accumulation is restricted, as it cannot access the pixels in neighboring areas above or below the area assigned to the stripe. Paths crossing stripes are cut and cost information cannot be distributed over them. The performance degradation is mitigated by increasing the context through an additional area above and below the stripe to provide context for the path accumulation (Fig. 3.7). As can be seen in the evaluation chapter, a small additional border area is sufficient to obtain results that are comparable with a classical non-striped algorithm approach. The thread-level parallelism using stripes is implemented using OpenMP[84]. OpenMP is a programming API for parallelization on shared-memory architectures. It parallelizes on thread level by special compiler directives, that declare code as suitable for parallel execution, e.g. for-loops.

The independence of the accumulation paths could be used for parallel calculation as well (similar to [33], see Fig. 2.7). This leads to an additional synchronization area at the calculation of the summed path costs or a greatly increased memory footprint. This variant was implemented with the OpenMP API as well. The parallelized version did not lead to any speedup compared to the non-parallelized one. This parallelization approach might be too fine grained for OpenMP, as synchronization is necessary at every pixel and disparity in the cost cube.

3.4.2 Disparity Compression

In the classical formulation, the SGM algorithm will be executed in a dense fashion over the complete disparity range. Depending on the application's needs, it might be only be necessary to calculate the matches at bigger disparities with a coarser grid. As depth is proportional to the inverse of disparity (see equation 2.1), one can sub-sample the bigger disparities, while staying above a certain depth uncertainty threshold (idea detailed in [33]). A simple and computationally effective strategy is to calculate every disparity value at a range from 0 to 63. The sub-sampling is started at 64 and every second or fourth value is calculated in the range up to 127. This way, in a typical automotive setup with 50 degrees field of view and a baseline of 22 cm, depth resolution is reduced only for depth values below 3 m.

Apart from the data cost calculation, all parts of the algorithm are agnostic to the minimized costs. The disparity space can be sub-sampled in some parts of the data cost computation. Data costs are compressed into one continuous disparity label volume and then the rest of the algorithm is applied to this reduced volume (Fig. 3.8). The algorithm now works on labels instead of disparities. It assigns labels that minimize the overall costs and that are seen as consistent and reliable during the selection step and the Left-right check. The resulting label maps $d'(x, y)$ are mapped to disparities by an efficient and simple function:

$$d(x,y) = \begin{cases} f + s \cdot (d'(x,y) - f), & \text{if } d'(x,y) > f \\ d'(x,y), & \text{otherwise} \end{cases} \tag{3.11}$$

where f is the disparity starting the sub-sampling and s is the sub-sampling step width. This transforms all sub-sampled label disparities to their true values. Even sub-pixel estimation can be performed before the mapping step. Since the input pictures are not scaled, the data cost has to be forgiving to slight offsets created by sub-sampling. If a structure belongs to an image region with disparity d_1 and during the sub-sampling this disparity is omitted, the data term should result in an optimum at either the upper or lower side of d_1. Furthermore, the SGM smoothness penalties are altered indirectly, such that pixels with a disparity in the sub-sampled range are free to have larger disparity variations.

The number of operations of the core algorithm is roughly linearly related to the number of checked disparities and the additional remapping function has little overhead. Therefore, sub-sampling in the disparity range in this way results in a noticeable speed

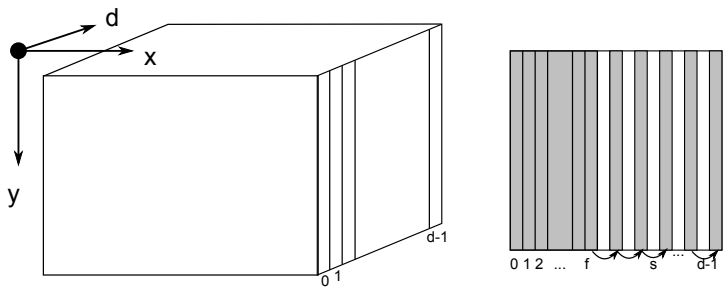


Figure 3.8: Disparity space compression: The uncompressed cost cube possesses one slice for each disparity value. During compression, the sub-sampling is started with a disparity f and a step width s and continues until the upper disparity $d-1$ is reached. Only the gray parts are part of the compressed slice. The white slices are skipped.

improvement. Of course, the reduction of disparity and thereby depth resolution must be acceptable for the application. In comparison to methods that carry out image sub-sampling, all image information is incorporated and the resulting disparity image itself is not scaled down. The angular resolution of the result is not reduced.

The algorithmic changes needed to implement this kind of sub-sampling are very small. The data cost calculation has to be replaced with a variant performing sub-sampling and has to add a remapping step at the end. The idea can be extended to multiple sub-sampling levels to get a less steep reduction of accuracy at the thresholds. However, the complexity of the matching cost computation increases as more cases have to be considered. Introducing more than two or three sub-sampling levels might create too much computational overhead.

3.4.3 Implementation Details

The Census-transformed images are calculated using SIMD instructions. For images with 8 bit intensity depth 16 pixels are processed in one step, as those fit neatly into the 128 bit wide SSE registers. For 12 bit intensity images, only 8 pixels can be evaluated at once. The 5×5 mask can be represented by thirty-two bit integers to efficiently save the result of the transformation. The 9×7 mask fits to 64 bit.

The cost computation step is executed $h \cdot w \cdot d$ times in each frame and uses a considerable amount of memory bandwidth, making it time-critical. Therefore, the calculation is parallelized on image stripes and uses SIMD instructions. The Hamming distance of two census values is calculated through an exclusive or (XOR) of these values followed by a population count of the result. The population count being the number of set bits in the result. This operation is the most time consuming part. The following implementation options are available:

- Lookup-tables
- specialized hardware instructions

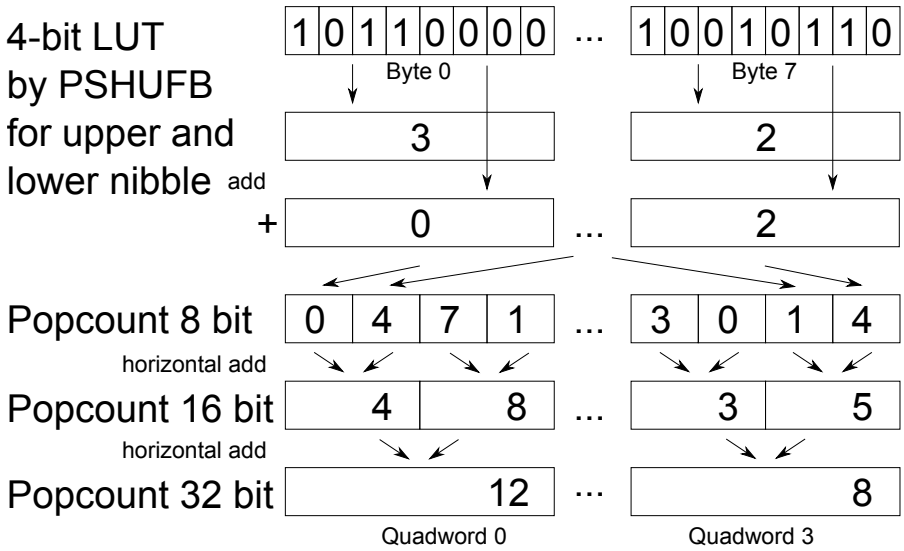


Figure 3.9: Calculation scheme: Parallel population count using hardware instruction based LUTs and horizontal additions.

- vectorizable logic
- hardware lookup-tables

Lookup-tables (LUTs) are the most efficient option without using commands from a special instruction set that might be available. Their disadvantage is that they cannot be parallelized and that they consume a sizable amount of the memory bandwidth and pollute the cache, as they easily need memory amounts comparable to an L1 or L2 cache size. As the whole value range must be covered, a LUT for n bit needs 2^n entries. That means that the upper size limit is 16 bit, requiring at least 64 kB for this size. Population counts for wider ranges must be created by multiple lookups for splits and added then up.

Hardware population counts are sometimes available as a special instruction (e.g. POPCNT). These are faster than LUTs, but usually only transform single values. Population counts are computable by logic operations as well, which can be parallelized easily, as presented by Humenberger in [55]. The Intel SSSE3 instruction PSHUFB enables the possibility to create a vectorized 4 bit-LUT in logic [79]. This command can be used together with horizontal additions to retrieve population counts for 32 and 64 bit wide integers (Fig. 3.9). No memory bandwidth is used in contrary to the lookup-table method and multiple values can be calculated in parallel, which compensates for the additional calculations. The main advantage of the hardware lookup-table solution is that the full SIMD register width can be processed. This matches better to the vectorized load and store paradigm and the other SIMD instructions used. Furthermore, it is more efficient than parallel logic, as the values for the nibbles can be retrieved by a simple instruction.

Path accumulation is performed using equation 2.11. It must be calculated $8 \cdot h \cdot w \cdot d$ times per image, when integrating 8 paths. As the aggregated costs are limited to a

16 bit value, 8 disparities can be processed in one step. The possibility to use special hardware instructions with saturation for addition, subtraction, maximum and minimum enables a speedup higher than 8, as it is possible to formulate the algorithm in a more direct way. The number of conditional jumps can be reduced by this approach as well.

The individual paths are integrated to the aggregated sum by grouping them into only two scans (Fig. 2.7). The downward first scan traverses from the top left of the image to the bottom right. Each row is completed fully before the next one is started. The first 4 paths are evaluated, saving an intermediate sum for each pixel disparity pair. The upward second scan traces back from the bottom right of the image to the top left incrementing the intermediate with the costs of the other 4 paths, resulting in the accumulated costs S.

A standard winner-takes-all search for the disparity with the minimum cost is performed. The second minimum is computed as well and is evaluated to invalidate the result, if the minimum is not sufficiently unique. This has to be executed for each of the $h \cdot w$ pixels of the reference disparity image (left side)and matching disparity image (right side). Using the SSE4 instruction PHMINPOSUW it is possible to calculate the minimum and its position in a vector of 8 cost values in parallel. Sub-pixel estimation is done at this step as well, as the cost values of the neighboring disparities are readily available. The minimum search of the right image is performed as a diagonal search in the disparity space image. For this part, the non-aligned memory layout prohibits direct fast SIMD processing. For each pixel location a copy of the costs for all disparities is created in a linear intermediate storage, which allows the disparity selection to be performed as was for the left image.

A 3×3 median filtering is performed in parallel without utilizing branches using SIMD instructions. A sorting network is implemented for this with parallel instructions for minimum and maximum. The last step is the left-right check.

In order to keep the overhead of the disparity space compression low, specific implementations for every variant have to be crafted. The data cost calculation is modified accordingly. The remapping function is implemented using SIMD instructions and without branches.

The experimental evaluations are located in subsection 6.2.3 in the evaluation chapter and give insights in quality and speed of the algorithm.

3.4.4 Striping in Space instead of Time

The striping approach cannot only be used to decrease runtime consumption. If memory is scarce or input images are big, the memory requirements of the normal SGM algorithm might by too high. If the striping is performed not in parallel, but in sequential fashion, the memory consumption can be reduced. Typically, the memory requirement of SGM is in $O(w \cdot h \cdot d)$, as the dominant memory structure is the disparity cost cube. If the data costs are computed in advance and re-used a full disparity cost cube is needed for the data costs and another one for the aggregated costs. With necessary bits for the data costs are denoted as b_d, the memory for the by far biggest data structures has the

following size in bits:

$$2 \cdot (b_d \cdot w \cdot h \cdot d) \tag{3.12}$$

For a full HD image of 1920 $\times$ 1080 pixels, 128 disparities and 16 bit cost depth the needed amount of memory is in the magnitude of 1 GByte. On embedded hardware this can be a limiting factor. With image striping the amount of used memory can be reduced to:

$$2\frac{b_d \cdot w \cdot h \cdot d}{n} \tag{3.13}$$

with n being the number of stripes. If, at the expense of increased runtime, the data costs are not computed in advance but on-the-fly, then the factor of 2 can be dropped, but the data costs have to calculated at least twice, at first for the downward pass and then for upward pass. If the upward pass is skipped, the determination of the best disparity can be done instantly and the cost cubes do not have to be stored. This algorithmic change comes at the price of reduced matching quality and non-symmetry of the results, as the upward paths are left out and no information can be spread in this direction.

4

Landmark Detection and Mapping

A reliable detection of time and viewer perspective landmarks is of crucial importance for a landmark-based localization scheme. Therefore, the selection of pole-like structures as a primary landmark is motivated and detailed. Detection is based on the evaluation of depth maps generated by stereo vision. Hypothesis generation works on detected depth edges and depth edge contours. Several properties for pole-like structures are derived and used in a classification step. There follows a description of the tracking process for detected landmarks using a Kalman filter and of how the mapping of landmark structures is handled.

4.1 Pole Extraction and Classification

The goal of this step is to extract pole-like structures in rural and urban environments. These can be trees, sign-posts, lighting poles or bollards (see Fig. 4.1). These objects are often present in these locations and have a very slow rate of change, enabling long-term mapping and localization. They differ rather strongly in appearance from the other common and non-static objects like vehicles, humans or trucks. Therefore, they are good landmarks. Other landmark candidates are planes e.g. at walls, building or window edges or characteristic surface changes like curbs. These could complement the approach to increase the availability. Thinking of other complementary features that do not possess a three-dimensional signature, road markings and delimiters are a valuable addition.

The pole detection process follows the following outline, being composed of a heuristic hypothesis generation combined with a classification step:

- depth edge detection on disparity maps
- contour finding using the detected edges
- grouping of contours to pole hypotheses

Figure 4.1: Urban scenarios: Often pole-like three-dimensional structures are present, with the additional feature being long-term stability of position and shape, like trees, bollards or poles of street lamps.

- attribute calculation for pole hypotheses
- classification

Working on disparity maps for the majority of the hypothesis creation steps enables the algorithm to make use of the density of this representation naturally. In point clouds, it is not clear how the next neighbor in the data structure relates to the current depth point.

Another approach to pole extraction would be to use a stixel world representation as an intermediate data structure (see section 2.2.2). A classification step would extract stixel groups with poles from the other objects. Possible drawbacks of such an approach are the loss of contour information for classification and the missing possibility to model branches and or twigs. This is only possible with one of the most advanced stixel representations [87], that requires a significant amount of processing power for its creation. Therefore, this can only be applied to scaled-down disparity images in a real-time context. A reduction of the detection range is therefore probable. In the context of a general object detection framework, it can be advisable to follow this approach, as this could be more resource efficient overall.

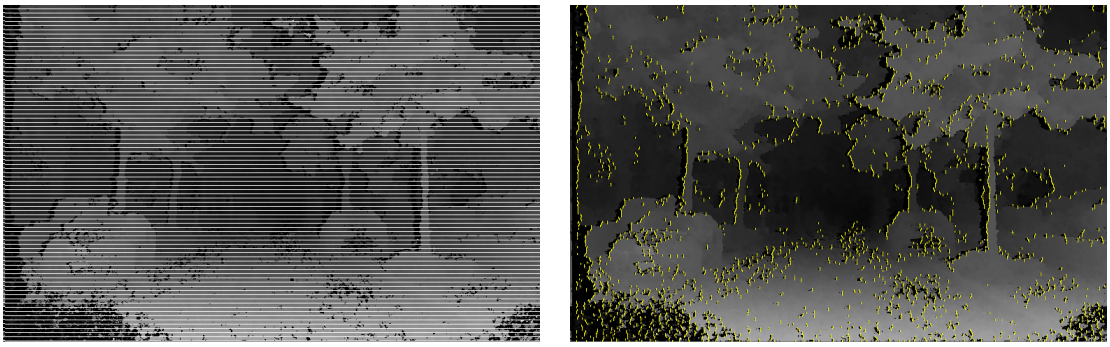
4.1.1 Depth Edge Detection

A horizontal Sobel operator of size 5×5 is applied to the left and right disparity image, that have been sub-pixel refined. The resulting gradient images are binarized with a dynamic threshold. Edges are detected not on the whole image, but on search-lines with a constant spacing. The dynamic threshold T_d for a disparity map d is calculated as a multiple of the standard deviation of the gradients in the current gradient image:

$$T_d = \alpha_T * \sigma(\text{Sobel}(d)) \tag{4.1}$$

This approach treats edges as outliers, as they are only present in a fraction of the whole image. This way, smaller gradients that represent depth clutter are filtered out and the threshold is automatically adapted to the scene.

Pixels in the gradient image on a scan line with an absolute gradient above the threshold are treated as possible edge locations. A non-maximum suppression reduces the



(a) Search lines (b) Depth edges

Figure 4.2: Edge detection in disparity maps: The detection is performed on scan lines (typical spacing 2 lines, spacing here 6). Rising and falling edges are detected based on dynamic thresholding. For clarity, only rising edges are shown here.

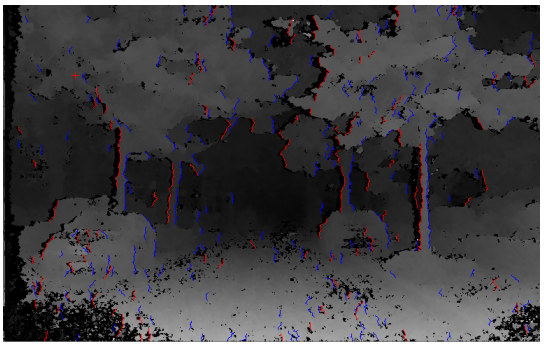
number of edges further, as it filters out edges that are surrounded by other ones with higher gradients. Edges can be of two types, rising edges at borders with increasing disparity and falling edges at borders with decreasing disparity (Fig. 4.2).

Occlusion effects are used to reliably detect the borders by using the left and right disparity images. This is related to the ideas developed in [16] that propose looking for left occlusions in the left depth image and right occlusions in the right image. Therefore, rising edges are detected in the left disparity image and falling edges in the right disparity image. Edges found in the right image are reprojected to the left image to form a combined edge structure. The output of this step is a list of locations of rising and falling edges with associated disparity and depth gradient values.

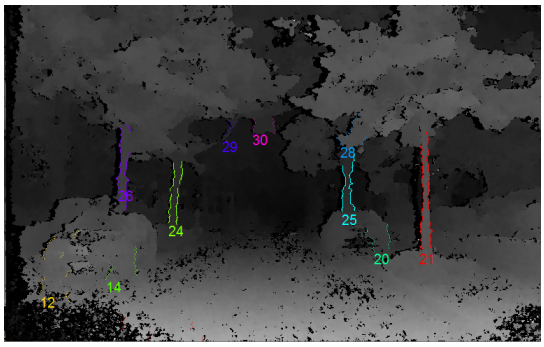
4.1.2 Contour Finding

The hypothesis generation for contour finding is a greedy algorithm, starting at the bottom of the image on the left side. It tries to form a contour by connecting edges that locally share the same direction. Therefore, it looks for the edge with minimal distance to the current contour on the following search lines. The search is constrained by a maximum allowed disparity change between adjacent edges. If no suitable edge is found on a search line, the scan is continued further up, until a maximum search range without added edges is reached. Every edge can only be part of one contour. If no edge can be added any more, the contour is finalized and the search for a new boundary continues to the right of the starting edge of the last contour. If the last edge was the rightmost edge of the scan line, the search is continued further up on the next one. The contour following is finished, if every edge has been visited and is either a part of a contour or has been checked as a possible start. The output of this step is a list of contours, that describe rising and falling boundaries in the disparity image. Figure 4.3a shows example results.

Contours are then filtered by their size and direction in the image. Ones that are too



(a) Contour finding results: Rising contours are red and falling contours are blue.



(b) Hypothesis generation: Generated hypotheses

Figure 4.3: Contours and pole candidates

small are rejected and directions that are too horizontal are excluded as well, as their probability to be part of a pole-like structure is low. The idea behind this is to remove contours created by edge clutter and by objects that are not mostly horizontal, as e.g. vehicles.

4.1.3 Hypothesis Generation

Contours are then merged into contour groups that form a pole candidate and ideally shall contain the majority of the object boundary of a pole. Contours are sorted by their starting position in the image. The merge process starts from the bottom of the image and continues from the left side to the right and then up. Starting with a rising contour on the left, one looks for a matching falling contour to the right of it that overlaps significantly, has a matching disparity and leads to a pole candidate width that is reasonable for the desired objects. This forms the initial pair of the contour group. The algorithm now looks for suitable enhancements at the upper part of it. These are spatially overlapping contours with an opposite direction that continue the detected shape. If a matching contour is found, it is added to the group and the search goes on until no enhancement to the group can be found. The segment group is finalized and the grouping continues with the next contour part still to be grouped.

A merge step after the initial creation handles situations with two pole candidates that have been created on one pole. This happens if there is no sufficient overlap between the contours that form both candidates. As they are located directly over each other and share the same depth, the merge step can be performed easily by a pairwise test of all created pole candidates. The search for mergeable candidates continues until no merge was performed in the last round of the algorithm. The output of this step is a list of contour groups that are hypotheses for pole-like structures (see Fig. 4.3b).

4.1.4 Pole Attributes and Classification

As the hypothesis generation creates a lot of candidates that do not actually belong to desired structures, a classification approach was pursued to filter out false positives.

The following feature classes were used for classification:

- *world dimension/geometry-based*: height, start height of the pole over ground level, end of pole over ground level, the lateral angle of the structure
- *combined features based on world dimensions*: height to width ratio, mean disparity value, x start position squared, height probability (Gaussian based), width probability (Gaussian based)
- *appearance-based features*: "completeness", a boolean flag indicating a very large height, "straightness", "disparity stability".

So apart from purely geometric properties like width and height and their ratios, appearance-based features are employed.

Feature Calculation

World dimensions are obtained by least-square fits. Width, height and lateral angle are estimated. The depth resolution does not admit the calculation of a longitudinal angle, especially for bigger detection ranges of more than 30 m. The assumption is that the angles of the structures will only be slight and that therefore the approximation in one direction will not be too detrimental. Furthermore, the viewing directions on a normal road will be mostly only from two sides. So the missing orientation angle will not lead to performance degradation due to changes in visibility.

The appearance-based features are computed as follows. "Completeness" is hereby defined as the number of depth edges that form the shape of hypothesis relative the maximum possible number of depth edges it could possess. Candidates with a few defining edges are more probable to be false positives, as their shape should not be so distinct. "Straightness" is measured by differences in lateral position along the candidate. Larger variations in these positions should induce a penalty. "Disparity stability" is a measure of how large the variation of the disparities belonging to the candidate is, measured along scan lines. Small variations should indicate a high quality matching. A local statistic along scan lines is employed, as the depth of e.g. trees might change with height, as they might be inclined.

Feature Selection

After an initial feature set was developed the features were selected by their rank by weight in logistic regression. Low ranks were discarded one-by-one and new features added the same way. If significant weight was attributed to the new feature, it was included in the feature set. The final selection of the semi-automatic process was validated using recursive feature elimination [40]. This process generates a feature ranking by applying the following procedure:

1. Train the classifier (optimize the classifier weights w_i with respect to the objective function).

2. Compute the ranking criterion (usually w_i^2).

3. Remove the feature with the smallest ranking criterion

Lowest ranked features were excluded to test whether they had a relevant influence on the classification metrics.

Further possible features are: the existence of branches and or twigs to favor trees, a semicircular shape or consistent intensity inside the contour. Local contour shapes seem to be not that stable, especially in the far region. Therefore, they were not used in the classification. However, they could be useful in the near region, to disambiguate between nearby poles. This is similar to features used for the classification of tree leaf shapes [24, 56].

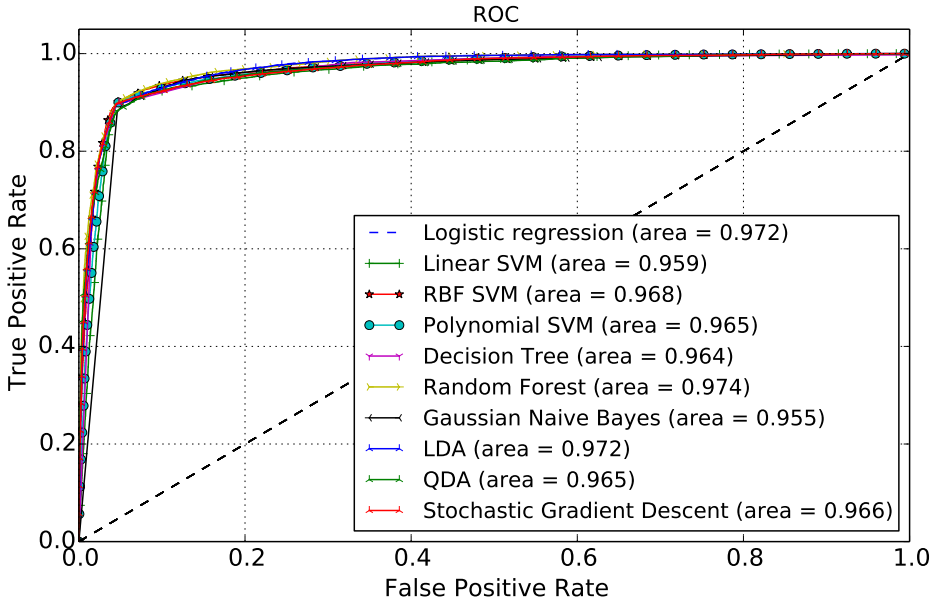
Classifier Selection

A test set was labeled manually to enable an offline training of the classifier. Two test scenes with a size greater than 2500 frames and 11000 positive and 19000 negative examples were used. All generated hypotheses were classified into three classes: pole, non-pole and unknown. Hypotheses labeled as unknown were excluded from the training process.

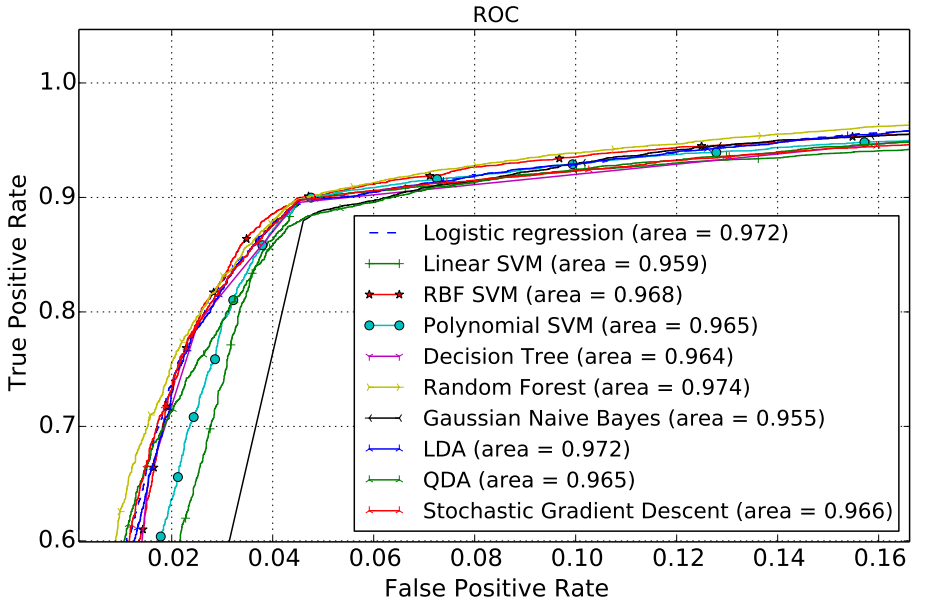
As a preparation step for classification, features are normalized to a mean of zero and standard deviation of one. Several standard classification approaches were tested with implementations provided in scikit-learn [85]: Logistic regression, Support Vector Machines (SVMs) with linear, polynomial and radial basis functions, Linear Discriminant Analysis (LDA) and its quadratic form (QDA), Naive Bayes classification - all covered by [2], decision trees [92] using an optimized version of the CART algorithm [20], Random forests [19], and Stochastic Gradient Descent (SGD) [17].

The classifiers were trained using a stratified 3-fold cross-validation [57]. The training set was created using 25% of the labeled data by stratified shuffling, the test set is formed by the remaining 75%. The resulting accuracies on the test set are detailed in the form of receiver-operator characteristics (Fig. 4.4). Area under the curve (AUC) is calculated as well as an overall measure of performance. The classifier quality is nearly equal for the majority of the approaches for bigger false positives rates. It only differs for the small ones. For clarity, a close-up of the "sweet spot" around 4% false positive rate is presented as well.

As logistic regression shows a small fall off in true positives with lower false positive rates, and the only method with a better AUC score is random forest, which is prone to over-fitting, I choose logistic regression as a classification method. Furthermore, the implementation of the classification step is straightforward and computationally lightweight.

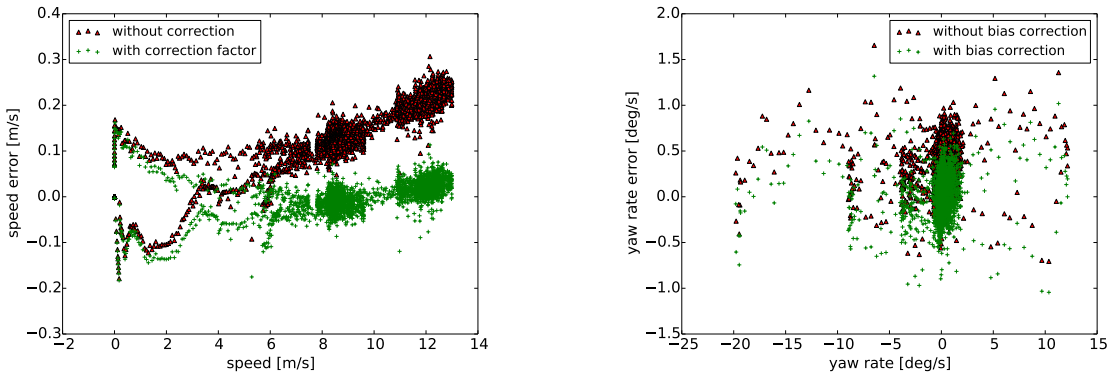


(a) full curve



(b) close-up

Figure 4.4: Receiver operator curve for different classifier types: The differences at high false positive rates are small. At a rate of 0.04 to 0.03, differences in performance between classifiers begin to appear. Logistic regression, radial basis function SVM and random Forests show the best true positive rate in this region.



(a) Speed error vs. speed

(b) Yaw rate error vs. yaw rate

Figure 4.5: Odometry error correction for wheel speeds and yaw rate: A scaling error for wheel speeds and a constant bias for yaw rate are compensated.

4.2 Odometry Offset Calculation

The vehicle transmits the speed signals of four wheel tick sensors and a yaw rate estimated by a gyroscopic device measuring the angular velocity of the vehicle along the Z axis. The wheel ticks are translated into wheel speeds by a multiplication with a model specific wheel circumference, which depends on tire pressure and wear. The major error source for the wheel ticks is an error in the wheel circumference, translating into a scaling error. This error is classified as long-term changing and short-term stable, therefore it was calibrated off-line once using the reference system. The yaw rate is subject to temperature dependent offsets of the gyroscope, leading to an additive sensor bias, which is likely to change rather fast, if the sensor temperature changes. An online calibration of the error is therefore favorable.

Using the Applanix INS/DGPS as a reference, the differences between the sensor readings and reference system were analyzed and the wheel speed scale error could be calibrated. Using this scaling factor, the mean error for the wheel speeds could be reduced from 0.104448 m/s to 0.001694 m/s with a standard deviation of 0.027337 m/s. At lower speeds, the resolution of the wheel tick sensors decreases, leading to a maximum absolute deviation of 0.144386 m/s (Fig. 4.5a). The estimated scaling factor was 0.985805. The decreased accuracy of the wheel ticks at very low speeds can be seen as well in the scatter plot.

The additive sensor bias for the yaw rate is calibrated by the following method. When the vehicle is stationary the yaw rate of the vehicle has to be zero. As the wheel tick sensors report the speed of the vehicle, vehicle standstill can be detected and the outputs of the yaw rate sensor can be integrated by summing them up. The resulting sum, divided by the number of sensor readings when stationary is the additive sensor bias. As the wheel ticks are not that accurate at very low velocities and they might report a velocity of zero even if the vehicle is moving, a delay of several seconds should be used

for before entering and before leaving the detected stationary state. Compared to the Applanix reference, the bias can be reduced from $-0.342726\,^\circ/s$ to $0.008134\,^\circ/s$ with a standard deviation of $0.228384\,^\circ/s$ (Fig. 4.5b). The figure shows the independence of the error on the vehicle turning rate, justifying the error model.

In urban scenarios, the frequency of re-calibration is sufficient, but for longer drives it might be necessary to think of an offset calculation that does not rely on the vehicle standing still. The wheel-ticks could be used for this, as they provide, through the differences of the wheel speeds, indirect information about the yaw rate. An integrated model using a Kalman filter seems necessary to model the noise characteristics of such a fusion approach.

4.3 Landmark Tracking

4.3.1 Coordinate Systems

Four different coordinate systems are used in the domain of landmark tracking and mapping (Fig. 4.6):

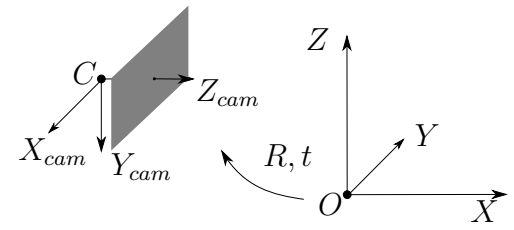
- the stereo camera coordinate system
- the vehicle coordinate system
- local East-North-Up coordinates (ENU)
- the World Geodetic System 1984 coordinate system (WGS84)

Measurements of the landmarks are done in the camera coordinate frame. A vehicle centered coordinate system with origin at the center of the front axle projected to the ground (Automotive Front Axle Coordinate system at the ground, AFAC-G) is used for landmark tracking. WGS84 is a geodetic reference system for standardized position description on the earth and in near-earth space. It comprises a reference ellipsoid, fitted optimal to the earth surface for location descriptions with longitude and latitude, a more detailed geoid to model deviations from the ellipsoid and coordinates of reference stations to relate both models to physical locations on the ground. WGS84-based coordinates with latitude, longitude and height are used to uniquely store the landmark coordinates during mapping. To enable an easier computation, the landmark locations are transformed into a local ENU frame during localization and for map searches. In this frame, a local reference point defines the origin, the x-axis is heading east, the y-axis is heading north and the z-axis is heading upwards. The units of all axes are meters.

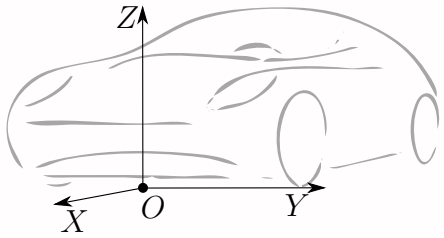
4.3.2 Kalman-Filter Design

The filtered pole state is composed of the position of the pole in the vehicle coordinate frame:

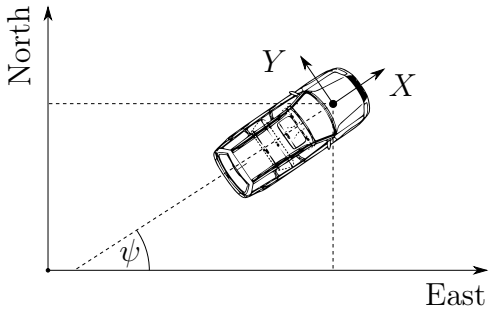
$$\mathbf{x}_t = (X, Y)^\top \tag{4.2}$$



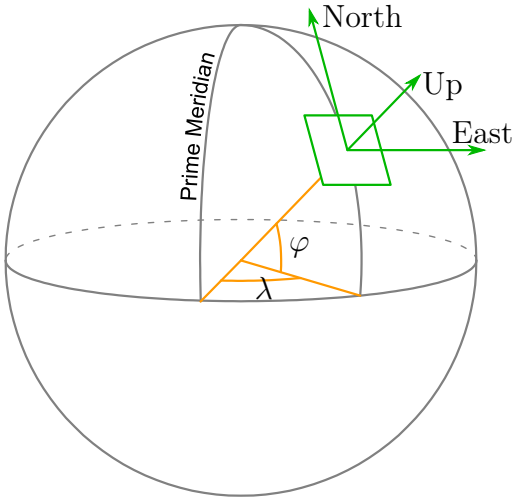
(a) Relationship between camera coordinate frame and vehicle coordinate frame



(b) Vehicle coordinate frame AFAC-G



(c) Relationship between ENU frame and vehicle coordinate frame



(d) Relationship between a geodetic reference system with latitude φ and longitude λ and an ENU frame

Figure 4.6: Coordinate frames for mapping and tracking: All four frames are needed in the pole mapping stage.

Other properties like width and height and the lateral angle ϕ are filtered with a mean filter. Measurements for the position are done in a translated camera coordinate frame $\mathbf{X}_{cam,\mathbf{t}}$, whose center is identical to the vehicle coordinate frame.

Prediction Step

The state is predicted by the available odometry information on vehicle speed v and yaw rate $\dot{\psi}$ over the time period Δt, using the axle-distance a for the axle-correction. They are transferred into changes of the vehicle coordinate system ΔX, ΔY and $\Delta\psi$ relative

to the global reference system:

$$\begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta\psi \end{pmatrix} = \begin{pmatrix} -\frac{v}{\dot{\psi}}\sin\psi + \frac{v}{\dot{\psi}}\sin\left(\psi + \dot{\psi}\Delta t\right) - a\cos\left(\psi\right) + a\cos\left(\psi + \dot{\psi}\Delta t\right) \\ \frac{v}{\dot{\psi}}\cos\psi - \frac{v}{\dot{\psi}}\cos(\psi + \dot{\psi}\Delta t) - a\sin\left(\psi\right) + a\sin\left(\psi + \dot{\psi}\Delta t\right) \\ \dot{\psi}\Delta t \end{pmatrix} \tag{4.3}$$

As the vehicle moves and the landmarks are tracked in vehicle coordinates, the inverse of this motion is included into the process matrices:

$$\bar{\mathbf{x}}_t = A_t\mathbf{x}_{t-1} + B_t\mathbf{u}_{t-1} \tag{4.4}$$

$$= R(\Delta\psi)^{-1}\mathbf{x}_{t-1} - I\begin{pmatrix} \Delta X \\ \Delta Y \end{pmatrix} \tag{4.5}$$

$R(\Delta\psi)$ being a rotation around Z.

Error Model for Stereo Measurements

Measurements of the position of the landmarks in the camera coordinate system rely on the disparity and location information of the landmark:

$$Z = \frac{b \cdot f}{d} \tag{4.6}$$

$$X = Z\frac{x_L}{f} = b\frac{c - c_0}{d} \tag{4.7}$$

$$Y = Z\frac{y_L}{f} = b\frac{r - r_0}{d} \tag{4.8}$$

The measurement noise is calculated using the commonly applied stereo error model with uncorrelated zero-mean Gaussians [71]:

$$\Sigma = \mathbf{J}_h diag(\sigma_c^2, \sigma_r^2, \sigma_d^2)\mathbf{J}_h^\top \tag{4.9}$$

, hereby $\mathbf{J}_h$ is the Jacobian of $h(c, r, d) \rightarrow (X, Y, Z)$ and $\sigma_c^2, \sigma_r^2, \sigma_d^2$ are the variances of the subscripted variables. In detail, this leads to:

$$\Sigma = \begin{pmatrix} \sigma_X^2 & \sigma_{XY} & \sigma_{XZ} \\ \sigma_{XY} & \sigma_Y^2 & \sigma_{YZ} \\ \sigma_{XZ} & \sigma_{XY} & \sigma_Z^2 \end{pmatrix} \tag{4.10}$$

$$= \begin{pmatrix} \frac{b^2\sigma_c^2}{d^2} + \frac{X^2\sigma_d^2}{d^2} & \frac{XY\sigma_d^2}{d^2} & \frac{XZ\sigma_d^2}{d^2} \\ \frac{XY\sigma_d^2}{d^2} & \frac{b^2\sigma_r^2}{d^2} + \frac{Y^2\sigma_d^2}{d^2} & \frac{YZ\sigma_d^2}{d^2} \\ \frac{XZ\sigma_d^2}{d^2} & \frac{YZ\sigma_d^2}{d^2} & \frac{Z^2\sigma_d^2}{d^2} \end{pmatrix} \tag{4.11}$$

Measurement Model

The relation between coordinates in the camera coordinate system and the vehicle coordinate system in homogeneous coordinates is

$$\mathbf{X}_{cam} = \begin{pmatrix} R & -R\tilde{C} \\ 0 & 1 \end{pmatrix} \mathbf{X} = \begin{pmatrix} R & \mathbf{t} \\ 0 & 1 \end{pmatrix} \mathbf{X} \tag{4.12}$$

, where $\tilde{C}$ is the camera location in the world coordinate frame. The relation between both coordinate systems is a rigid body transformation with a rotation R and a translation $\mathbf{t}$. The measurement equation maps a state $\mathbf{x}_t$ to a measurement $\mathbf{z}_t$ in the coordinate frame $\mathbf{X}_{cam,\mathbf{t}}$:

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t \tag{4.13}$$

with

$$\mathbf{H} = R^{-1} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \tag{4.14}$$

Update Step and Lifetime Management

As multiple landmarks are tracked at the same time, lifetime management and association of multiple measurements to multiple landmarks is necessary. Association is done by a greedy-assignment based on the Mahalanobis distance between measurements and state. The update step includes a validation gating check which as based on the Chi-Square distance [7]. This is a standard approach for Kalman filtering to increase robustness against outliers (see subsection 5.3.4 for gating in localization, where the approach is detailed further).

Landmarks with no recent measurements are deleted as are landmarks that have left the area observed by the camera. Not-assigned hypotheses create new landmarks. The measurement noise is also used as initialization for the state covariance during the setup of these new landmarks.

The experimental evaluation and field test results for landmark detection and tracking are located in section 6.4 in the evaluation chapter.

4.4 Landmark Mapping

Landmarks are considered for storage in the map if they have a maximum distance from the vehicle and have either left the camera field of view or have not been seen for a couple of frames. A precondition for them to be included in the map is that they should have been tracked for a minimal number of frames during their lifetime. This is done to reduce the fraction of false positives and unstable poles in the map.

In order to have a locally smooth and globally precise location of the landmark, for mapping purposes, the localization results of the INS/GPS Applanix are used (see Fig. 4.7). Despite the repeatability problems and local drift, the accuracy of the derived

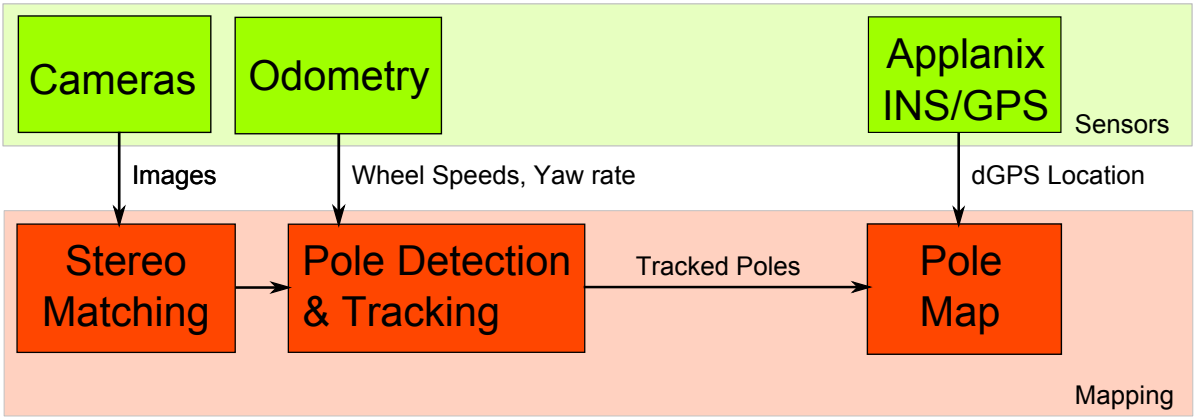


Figure 4.7: Mapping dataflow

map is sufficient for localization. Problems might occur at loop closures and with the connection of maps that were created at different times. This could be corrected by explicit modeling of the uncertainty of the Applanix, detection of loop-closures and global correction of the GPS offsets and drifts, as was proposed with GPS SAM [23].

Landmarks are stored in a kd-Tree with two dimensions for the landmark position. Additional attributes as pole width, lateral angle and metadata as a unique id and tracking statistics are archived as well. For the addition of the pole to the tree poles nearby are also queried. If a pole in the map is found near the new pole to be inserted, at a euclidean distance smaller than d_m, the two poles are merged to a new estimate.

In order to facilitate initializations, the mapping trajectory is also saved along the map. Mapping results are presented in section 6.3 of the evaluation chapter.

5

Localization for Autonomous Vehicles

As the term localization is quite broad in its usage, the following introduction gives an overview of the field and describes the goals of the localization solution developed in this chapter. There follows an explanation of localization by particle filtering, covering the state space, process model, likelihood and re-sampling scheme. As the localization output has to be updated for the vehicle controller with 100 Hz and should be smooth, an output Kalman filter is used to provide this update rate. The chapter closes with a system overview of the localization solution, including the deployment of the different parts.

5.1 Introduction

We follow the taxonomy of localization problems for robots as described in [113], separating the problem class into four dimensions:

- *Local vs. global*: The first category is separated by the available knowledge of the robot's position during the course of the operation: position tracking, global localization and kidnapped robot. While position tracking has information about the initial robot pose, this pose is unknown in global localization. The kidnapped robot problem is an even more difficult variant of global localization, where the robot can be kidnapped and taken to another position at all times without this being apparent.

- *Static vs. dynamic environments*: In static environments, only the robot moves, while in dynamic environments, other parts of the scene might change their position, e.g. doors, people or other vehicles.

- *Active vs. passive*: The localization can furthermore be either active or passive, meaning whether the localization has an impact on the control of the robot, with the aim to facilitate the localization. Active localization controls the robot with

the goal to improve the localization result, while passive localization just provides information on the position relevant to other robot components.

- *Single vs. multiple robot*: The last dimension considers whether a single robot is accomplishing the localization task or multiple robots together.

The implemented localization approach is able to determine itself globally, using GPS for initialization. The kidnapped robot problem is not tackled for unlimited teleportation, while small displacements of up to several meters can be handled in principle. The operation environment of the vehicle is dynamic, as other traffic participants are part of the scenario. Even the landmarks can change over a longer time span or at least be hidden due to occlusion effects. The localization is passive, as it only provides its result and does not influence the vehicle control directly and is performed by one robot alone.

5.2 Localization Particle Filter

We implement a Markov-Localization scheme using Monte-Carlo Localization. The position and orientation of the vehicle are estimated by means of a particle filter [28]. This is done to handle non-linear models for the motion of the vehicle and the likelihood. At the same time this offers the possibility to represent potentially multi-modal densities. A set of M particles $\mathcal{P} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)}\}$ with their corresponding weights $\mathcal{W} = \{w^{(1)}, \ldots, w^{(M)}\}$ represents the localization state as a probability density at each time step k according to

$$p(\mathbf{x}_k|\mathbf{Z}_{1:k}, \mathbf{M}) \approx \sum_{i=1}^{M} w_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}), \tag{5.1}$$

with $\mathbf{Z}_{1:k}$ being the set of all measurements from step 1 to k, $\mathbf{M}$ the landmark map and $\delta(\cdot)$ the Dirac delta function.

5.2.1 Localization State

Each particle $\mathbf{x}_k^{(i)}$ has a three-dimensional state $\mathbf{x}$ with UTM-coordinates Easting E, Northing N and heading ψ, relative to a fixed global zero:

$$\mathbf{x} = (E, N, \psi)^\top \tag{5.2}$$

This state represents the position and orientation of the vehicle in two dimensions.

5.2.2 Initialization

For particle initialization, the position and orientation of a low-cost GPS is used. Samples from a Gaussian around this pose with position variance σ_p^2 and heading variance σ_ψ^2

are created. The position variance is hereby derived from the GPS position uncertainty measure dilution of precision ([27], p.83ff).

The position of the GPS can be improved by searching for the nearest point on the mapping trajectory relative to the GPS position and using this point as the center for the position sampling. The idea is that the trajectory of the car during mapping provides an additional prior on the vehicle pose.

5.2.3 Localization Process Model

The process model or state transition distribution defines the probabilistic relationship between the past state $\mathbf{x}_{k-1}$, the past controls $\mathbf{u}_{k-1}$ and the current state:

$$p(\mathbf{x}_k|\mathbf{u}_k, \mathbf{x}_{k-1}) \tag{5.3}$$

This distribution is sampled to predict the particles in the course of the particle filter algorithm. The process model uses a bicycle model with constant velocity and front-axle correction (Fig. 5.1) to predict vehicle motion. In comparison to the standard model, our vehicle coordinate system is located at the front axle, while the vehicle moves round a circle connected to the rear axle. This is taken care of by additional terms providing the front-axle correction. Odometry sensors are modeled as control inputs with additional errors:

$$\mathbf{u}_k = (\widetilde{v_k}, \widetilde{\dot{\psi}_k}, \widetilde{\gamma})^\top \tag{5.4}$$

with measured speed $\widetilde{v_k}$, measured yaw rate $\widetilde{\dot{\psi}_k}$ and an additional final rotation $\widetilde{\gamma}$. The prediction is done over the time difference Δt between $k+1$ and k and the axle-distance is a (see [37], p.195 ff):

$$\Delta\psi_k = \widetilde{\dot{\psi}_k} \cdot \Delta t \tag{5.5}$$

$$\psi_{k+1} = \psi_k + \Delta\psi_k + \widetilde{\gamma} \tag{5.6}$$

$$E_{k+1} = E_k + \frac{\widetilde{v_k}}{\widetilde{\dot{\psi}_k}} \cdot [\sin(\psi_k + \Delta\psi_k) - \sin(\psi_k)] - a\cos(\psi_k) + a\cos(\psi_{k+1}) \tag{5.7}$$

$$N_{k+1} = N_k + \frac{\widetilde{v_k}}{\widetilde{\dot{\psi}_k}} \cdot [\cos(\psi_k) - \cos(\psi_k + \Delta\psi_k)] - a\sin(\psi_k) + a\sin(\psi_{k+1}) \tag{5.8}$$

The complexity of the error model is based on the odometry measurements. The assumption is now that the actual velocities differ from the measured ones by zero-mean errors with a defined variance:

$$\widetilde{v} = v + \varepsilon_{\alpha_1} \tag{5.9}$$

$$\widetilde{\dot{\psi}} = \dot{\psi} + \varepsilon_{\alpha_2} \tag{5.10}$$

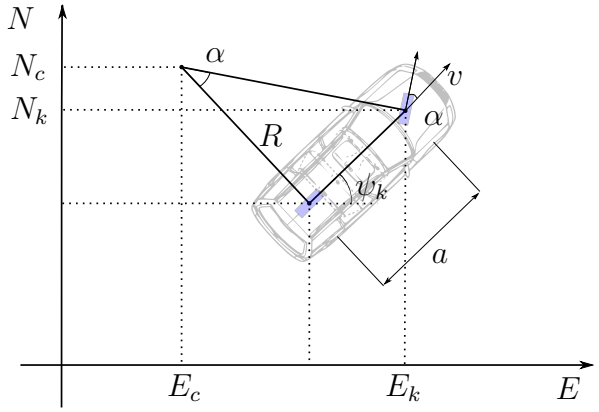


Figure 5.1: Bicycle motion model with front-axle correction: In contrast to the typical model, our vehicle coordinate system is at the front-axle, inducing an additional correction term.

Here ε_σ is a zero-mean error variable with standard deviation σ. The calibration values for the odometry errors are used here as well (see 4.2). Speed scale errors and yaw rate drift are reduced to a minimum and the error models for yaw rate and speed could be simplified to constant variances. In contrast to that, the final rotation error depends on the rotation speed of the vehicle, as rotation errors increase with turn rate:

$$\widetilde{\gamma} = \varepsilon_{\min(\alpha_3\dot{\psi},\alpha_4)} \tag{5.11}$$

The reason for the introduction of an additional final rotation, which is not justified by a physical equivalent, is that at least three noise variables are needed to sample the three-dimensional pose space without degeneracy (see [113],p.129).

5.2.4 Measurement Model

As the perception distance of the stereo rig is limited, the computation of the likelihood is limited to a region around the current best estimate. Landmarks stored in the map around the vehicle are extracted from the tree structure and associated to measurements found in the same region of interest. The chosen approach is similar to the method employed by Wiest et al. in [119], but adapted to the changed primary sensor with a different error model and changed clutter handling. Another option would be the omission of an explicit association step, as pursued for example by Lundgren et al. [68] for radar targets. However, since the fraction of clutter objects is not as high as with radar, explicit association was favored.

The likelihood is based on an optimal association between currently tracked poles, which constitute the measurements $\mathbf{Z} = \{z^1, \ldots, z^{z_k}\}$ (Fig. 5.2b) and the landmarks in the map near the current position $\mathbf{M} \supseteq \mathbf{M}_{near} = \{m^1, \ldots, m^{m_k}\}$ (Fig. 5.2a). As the landmarks are defined in a global ENU coordinate system and the measurements are defined in the vehicle coordinate system, the measurements are transformed into global

coordinates, using the particles' $\mathbf{x}_k^{(i)}$ position and orientation. The optimal mapping between those two sets $\theta : \{1, \ldots, m_k\} \rightarrow \{0, 1, \ldots, z_k\}$ is found by solving an optimal assignment problem (Fig. 5.2c). The assignment cost matrix is made symmetric by introducing clutter poles. The distance $\mathrm{d}(z, m)$ between a pole in the map m and a currently observed pole z is hereby defined as a combination of the Mahalanobis distance of their positions, using the position covariance S_z of the measurement (see Fig. 5.2b) and the difference in their diameters d_w, weighted by the diameter standard deviation σ_w:

$$\mathrm{d}(z, m) = \beta_p \cdot \mathrm{d}_M(z, m)^\top S_z^{-1} \, \mathrm{d}_M(z, m) + \frac{d_w^2}{\sigma_w^2} \tag{5.12}$$

The additional factor β_p is used to balance the influence of position and pole width. $\mathrm{d}_M(z, m)$ is the Euclidean distance of z and m. As landmarks in the map can be missed due to changes since the creation of the map and due to occlusions, assignments to clutter poles are included in the optimal assignment problem. This approach copes with false positive measurements as well. The covariances in the distance lead to solutions that include the stereo measurement uncertainty in the decision. Fig. 5.2c shows that possible ambiguities, as they appear with one tracked pole and two nearby landmarks, are resolved using the Mahalanobis distance.

The solution to the assignment problem is found by applying the Jonker-Volgenant algorithm (LAPJV,[59]), as it is usually faster than the classical approach, the Munkres algorithm [80]. However, due to numerical instabilities, LAPJV does not always brings forth to a solution. These cases are detected and the Munkres algorithm is used as a fall-back. Formally, the optimal assignment can be represented as the function, that associates landmarks with measurements

$$\theta : \{1, \ldots, \hat{m}_k\} \rightarrow \{0, 1, \ldots, \hat{z}_k\} \tag{5.13}$$

The first measurement 0 is hereby a clutter measurement and each real measurement can be associated to not more than one landmark:

$$\theta(i) = \theta(j) > 0 \rightarrow i = j. \tag{5.14}$$

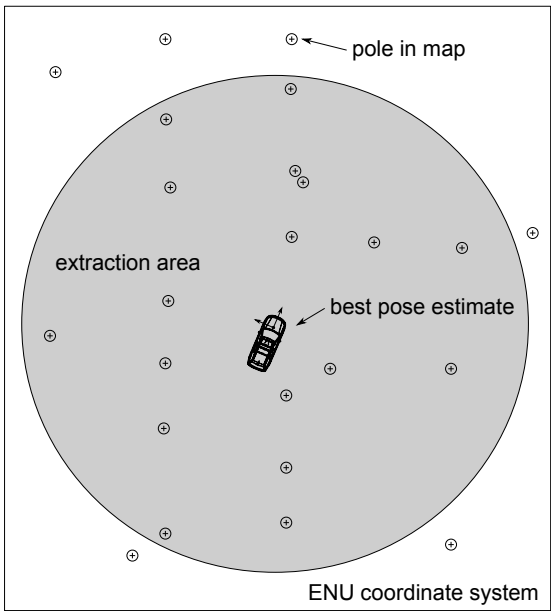
The likelihood $\mathrm{g}(z^j|m^l)$ for the assignment of landmark m^l to measurement z^j is given by

$$\mathrm{g}(z^j|m^l) = \frac{p_D}{\kappa(z^j)} \exp(-\frac{1}{2}(d(z^j, m^l)), \tag{5.15}$$

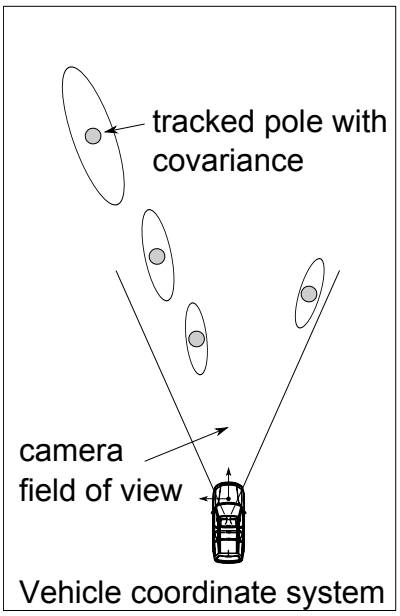
with $\kappa(z^j)$ being the intensity of a Poisson distributed clutter process (see [116] for an in depth derivation of the approach) and p_D the detection probability of a landmark. The likelihood $\mathrm{g}(z^0|m^l)$ for a not-detected landmark m^l is defined as:

$$\mathrm{g}(z^0|m^l) = 1 - p_D \tag{5.16}$$

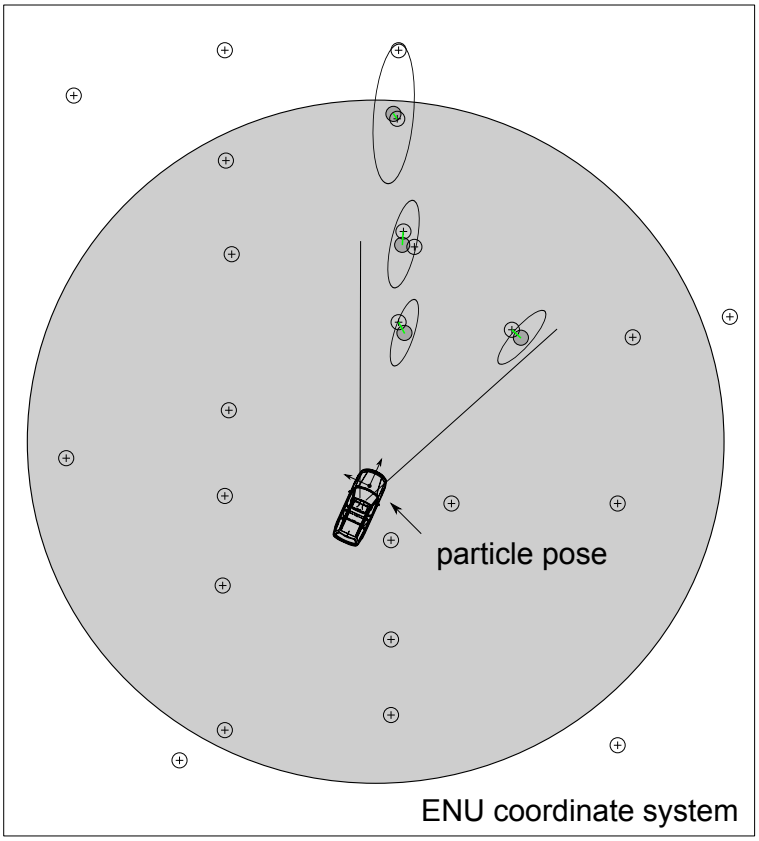
The overall likelihood w_k^i for the particle $\mathbf{x}_k^{(i)}$ is the product of the likelihood of all



(a) Extracted landmark poles from map using particle filter best pose estimate



(b) Pole measurements with position covariances



(c) Optimal assignment given the particle position and orientation

Figure 5.2: Likelihood computation process for a particle position and orientation

measured poles with its optimal association θ:

$$w_k^{(i)} = p(\mathbf{z}_k|\mathbf{x}_k^{(i)}, \mathbf{M}) = \prod_{l=1}^{m_k} \mathrm{g}(z^{\theta(l)}|m^l) \tag{5.17}$$

5.2.5 Resampling Strategy, Exploration and State Estimation

The goal of resampling is to control the sample variance of the particles representing the probability density. Continuous resampling leads to a loss of diversity and reduces the approximation accuracy, while never resampling may lead to a very ineffective sampling of the density, as the majority of the particles reside in low probability areas. With this in mind, the Low Variance sampler [113] is applied to resample the particles, only if a degenerated particle distribution is detected. This is accomplished by calculating a measure for the variance of the particle weights. One possible measure is the number of effective particles N_{eff}. Assuming that the weight of the particles is normalized to be a sum of 1, N_{eff} is defined as

$$N_{eff} = \left(\sum_{i=1}^{M} (w^{(i)})^2 \right)^{-1} . \tag{5.18}$$

Smaller values of N_{eff} mean a higher variance of the weights, the higher, the more degeneracy. In this case only a small fraction of particles have high weights, while the other ones have only very small weights. If the number of effective particles N_{eff} drops below a threshold $T_{N_{eff}}$, degeneracy is detected.

In order to increase the robustness of the filter, a small fraction of particles can be placed uniformly in the state space and the density at these positions can be explored. This is called active exploration. As this distorts the true probability density, it should only be done if performance indicators of the filter indicate a loss in confidence. Active exploration takes place, if the short-term filtered normalized likelihood drops in relation to the long-term filtered normalized likelihood. This approach is a variant of Augmented MCL (proposed in [113], p.256ff). Normalization is done by excluding matched clutter poles and taking the k-th root of the likelihood, if it was created by k matched poles. The idea is to create a likelihood value that is less influenced by changes in the map and more by the uncertainty of the associations. If the exploration condition becomes true, a small fraction of particles with low weights are re-initialized from a random uniform density centered around the current best estimate.

The localization result of the particle filter $\mathbf{x}_p$ is extracted from the particle set by taking the arithmetic mean of the values of all particles. A full covariance matrix Σ_p is extracted as well to measure the uncertainty of the filter. This simple Gaussian approximation $(\mathbf{x}_p, \mathbf{\Sigma}_p)$ only provides good estimates in unimodal distributions. More complex and robust approaches like k-means clustering, density trees or Mean-Shift [25] were not pursued as the localization quality is sufficient even with this simple approach. As the result is not directly used, but filtered by the Output Kalman filter, additional

robustness is provided later in the localization system.

5.3 Output Kalman Filter

One of the most important requirements that the output Kalman filter has to fulfill is an output rate of 100 Hz for the vehicle position, heading, speed and turn rate with minimal possible latency, as the vehicle controller works at this update rate [38]. Furthermore, the resulting trajectory should be smooth, with only small jumps in localization and it should be possible to compensate or at least hide the delay that is introduced to the pose result by the processing time.

The requirements are met by the following Extended Kalman filter (EKF), which integrates the particle filter location and orientation results with odometry measurements from the vehicle sensors. It has a state with position, orientation, speed and yaw rate. A guarantee for smoothness of the trajectory is possible by measurement gating. No position/orientation change takes place at vehicle standstill, e.g. at traffic lights, which keeps the estimated covariances within bounds. The output rate is achieved by continuous integration of the available measurements and a short additional prediction. Using an Unscented Kalman Filter (UKF, [60]) instead of the EKF might be an option as well, but as its advantages for linear measurement models seem limited (see [100]), this approach was not pursued.

5.3.1 State and Process Model

As for the particle filter, a Constant Turn Rate and Velocity Model (CTRV) with front-axle correction is used. This is a bicycle model with the assumption of constant vehicle speed and yaw rate. The state has the following structure:

$$\mathbf{x} = (x, y, \psi, v, \dot{\psi})^\top \tag{5.19}$$

, with x, y, ψ describing the vehicle position in ENU. More complex models like Constant Turn Rate and Acceleration (CTRA), which at the same time models acceleration, are possible choices. Their advantages, however, are only revealed during heavy accelerations and decelerations [101], a specific application that rarely happens during autonomous driving. As the pose input from the particle filter is already filtered, the state model neglects the correlation between these inputs. This could be extended by state vector augmentation to model the colored noise [61]. This would lead to slightly more realistic variances, as variances based on correlated measurements tend to be too optimistic in theory, which is confirmed by a study on the filtering of GPS fixes [93].

With the resulting prediction function f the vehicle follows a circle with constant radius for a prediction time Δt. a is the axle-distance. This function is only defined, if the yaw rate $\dot{\psi}$ is not equal zero:

$$f(\mathbf{x}) = \begin{pmatrix} x + \frac{v}{\dot{\psi}}\left(-\sin(\psi) + \sin\left(\Delta t\dot{\psi} + \psi\right)\right) - a\cos(\psi) + a\cos\left(\Delta t\dot{\psi} + \psi\right) \\ y + \frac{v}{\dot{\psi}}\left(\cos(\psi) - \cos\left(\Delta t\dot{\psi} + \psi\right)\right) - a\sin(\psi) + a\sin\left(\Delta t\dot{\psi} + \psi\right) \\ \Delta t\dot{\psi} + \psi \\ v \\ \dot{\psi} \end{pmatrix} \tag{5.20}$$

The Jacobian of this prediction function J_f at $\mathbf{x}$ is:

$$\begin{pmatrix} 1 & 0 & a\sin(\psi) - a\sin(\psi') + \frac{v}{\dot{\psi}}(-\cos(\psi) + \cos(\psi')) & \frac{1}{\dot{\psi}}(-\sin(\psi) + \sin(\psi')) & -\Delta t a\sin(\psi') + \frac{\Delta t v}{\dot{\psi}}\cos(\psi') - \frac{v}{\dot{\psi}^2}(-\sin(\psi) + \sin(\psi')) \\ 0 & 1 & -a\cos(\psi) + a\cos(\psi') + \frac{v}{\dot{\psi}}(-\sin(\psi) + \sin(\psi')) & \frac{1}{\dot{\psi}}(\cos(\psi) - \cos(\psi')) & \Delta t a\cos(\psi') + \frac{\Delta t v}{\dot{\psi}}\sin(\psi') - \frac{v}{\dot{\psi}^2}(\cos(\psi) - \cos(\psi')) \\ 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

with $\psi' = \Delta t\dot{\psi} + \psi$. If the yaw rate is zero, the vehicle just travels straight on for $\Delta t v$ in the direction ψ instead of turning on the circle. In this case the resulting Jacobian is simplified accordingly and used in the filter steps.

5.3.2 Noise Modeling

The process noise matrix $\mathbf{Q}$ defines the way the system uncertainty increases if the process is run in an open loop without measurements for a time span Δt. No clear design rules exist for this matrix. One popular approach is to fill only the diagonal, based on errors due to not modeled system dynamics, e.g. accelerations:

$$\mathbf{Q} = diag(\sigma_{pos}^2, \sigma_{pos}^2, \sigma_{\psi}^2, \sigma_v^2, \sigma_{\dot{\psi}}^2) \tag{5.21}$$

As we model car motion in the linear case, a maximum acceleration a_{max} of

$$a_{max} = 0.7g \tag{5.22}$$

is reasonable. With this assumption the result is:

$$\sigma_{pos} = \frac{1}{2} a_{max} \Delta t^2 \tag{5.23}$$

The heading variance is modeled as if no yaw rate is part of the process model:

$$\sigma_{\psi} = 30^\circ / s \Delta t \tag{5.24}$$

One could also opt to model only yaw acceleration errors, as the yaw rate is included in the state, which would lower this variance. The speed variance is derived from a_{max}:

$$\sigma_v = a_{max} \Delta t \tag{5.25}$$

The yaw rate error model is derived from assuming a maximum change rate of $20°/s^2$ which is justified from the analysis of data from u-turns in urban scenarios:

$$\sigma_{\dot{\psi}} = 20°/s^2 \Delta t \tag{5.26}$$

A more complex possibility for the creation of $\mathbf{Q}$ is the adoption of a Piecewise White Noise Model [8], which would fill some off-diagonal elements in $\mathbf{Q}$ as well. However, the absolute values of these elements are often very small, so that an omission of them often makes little difference in practice. Another approach is the additional restriction of position uncertainty by the use of the non-holonomic constraints of the vehicle motion. One could describe the uncertainties in velocity and yaw rate and then map them to the position uncertainty by an additional transform [61].

The measurement functions h for odometry and particle filter position inputs are direct mappings from measurements to state. In addition, we have to model the measurement noise matrices. $\mathbf{R}_o$ for odometry is a diagonal matrix with fixed $\sigma^2_{o,v}$ and $\sigma^2_{o,\dot{\psi}}$:

$$\mathbf{R}_o = diag(\sigma^2_{o,v}, \sigma^2_{o,\dot{\psi}}) \tag{5.27}$$

$\mathbf{R}_p$ for the particle filter position uses the scaled covariance matrix of the position estimate:

$$\mathbf{R}_p = \gamma_p \mathbf{\Sigma}_p \tag{5.28}$$

5.3.3 Measurement Integration and Output Generation

The 100 Hz output is generated by continuous integration of incoming position and odometry updates and intermediate prediction (Fig. 5.3). Odometry information can be included instantly and is available with an update rate of 50 Hz. Only signal offsets have to be corrected, thus processing delay is negligible. As the position updates depend on the particle filter and the rest of the localization pipeline, especially the stereo disparity computation, notable processing time is needed for the associated computations. This induces latency relative to the measurement time. Fig. 5.3 shows this schematically by stretching the output in time until it is available as input. Due to the dependency of the localization to the image input, synchronization gaps appear which further reduce the average update rate of the particle filter result. The Kalman filter does not wait for the localization result to be available but includes it, if it is available. It is included at the correct position in time, as the particle filter pose is time stamped with the stamp of the image, it is based on. The filter is rolled back in time for the position measurement and then rolled forward using all odometry updates that have arrived in the meantime. This hides the computation latency and minimizes the necessary prediction.

5.3.4 Validation Gating

The idea of a validation gate during the Kalman innovation step comes from the knowledge about the probability of a measurement, given the current filter state and mea-

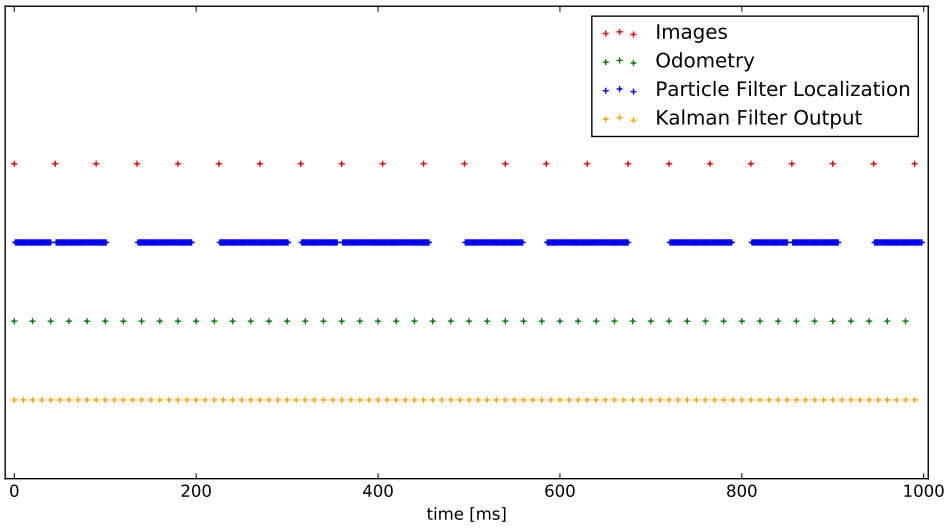


Figure 5.3: Schematic time diagram of measurement updates and then filter output: Odometry and localization measurements arrive asynchronously at the Kalman filter and are included, as they are available. The results of the particle filter are available after the processing delay and have the time stamp of the images, they are based on. The filter runs in open loop if no measurements are available.

surement uncertainty (see [8],p.263). Highly improbable measurements can be rejected, if one can measure this probability. A short review of the Kalman equation for an Extended Kalman Filter to define the measurement residual $\mathbf{v}$ and the measurement prediction covariance $\mathbf{S}$ follows. The prediction step equations are:

$$\mathbf{x}^- = f(\mathbf{x})$$

$$\mathbf{P}^- = \mathbf{J}_f \mathbf{P} \mathbf{J}_f^\top + \mathbf{Q}$$

The measurement step equations can be defined as:

$$\mathbf{K} = \mathbf{P}^- \mathbf{J}_h^\top (\underbrace{\mathbf{J}_h \mathbf{P}^- \mathbf{J}_h^\top + \mathbf{R}}_{\mathbf{S}})^{-1}$$

$$\mathbf{x}^+ = \mathbf{x}^- + \mathbf{K}(\underbrace{\mathbf{z} - h(\mathbf{x}^-)}_{\mathbf{v}})$$

$$\mathbf{P}^+ = \mathbf{P}^- (\mathbf{I} - \mathbf{K}\mathbf{J}_h)$$

One can now look at the normalized error e^2 and impose a threshold e^2_{max} on it:

$$e^2 = \mathbf{v}^\top \mathbf{S}^{-1} \mathbf{v} < e^2_{max} \tag{5.29}$$

Measurements are only included in the measurement step, if they stay below the threshold, otherwise they are ignored. The normalized error e^2 varies as a Chi-Squared distribution with the same degrees of freedom as the dimension of the measurement (see [8],p.236 for an in depth explanation). e^2_{max} is chosen to depend on the desired confidence level, e.g. 99%. This method allows the rejection of spurious erroneous measurements, depending on the filter uncertainty.

A practical example will clarify the dimensions and give a hint to the parametrization of e^2_{max}. Assuming a Kalman filter state with the following predicted process covariance matrix $\mathbf{P}_g^-$ and a position measurement with measurement covariance $\mathbf{R}_g$ (physical units in this example are meters for positions and radians for orientations):

$$\mathbf{P}_g^- = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 10^{-5} \end{pmatrix}, \mathbf{R}_g = \begin{pmatrix} 0.05 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 0.003 \end{pmatrix} \tag{5.30}$$

and a confidence level of 0.999 leading to a threshold e^2_{max} of 16.266. Then one example border case for the residual, which just passes the validation gate, is a measurement with residual $\mathbf{v}_g = (-0.697, -0.697, 0.0047)^\top$ and a e_g^2 of 16.2649. This measurement leads to a change in position and orientation due to the measurement of

$$\mathbf{Kv} = (-0.11616667, -0.11616667, 0.00015161)^\top.$$

The jump in the filter output is therefore slightly below 17 cm, which is tolerated by the controller, but might show in a slight jerk at the steering wheel.

5.3.5 Zero Velocity Updates

Near zero velocities in the dimension of centimeters per second for a longer period are unreasonable for terrestrial vehicles and can be considered zero. A land-based vehicle does not move, if its speed is zero, therefore its location cannot change. If the velocity measurements are near zero for at least a short period e.g. several seconds, the filter switches into the zero velocity mode and does not update nor predict the state, until the measured vehicle speed differs distinctively from zero. This is done to prevent localization changes at vehicle standstill, e.g. a slow rotation in drift. Furthermore, the localization measurements would be now heavily correlated and would lead to an over-confident filter, if measurements are present or to a completely under-confident filter, if no localization measurements are created by the particle filter.

5.4 System Overview

The complete dataflow of the localization algorithm is depicted in Fig. 5.4. While the camera image is only used for the disparity computation, odometry information is used for pole tracking, particle filter and the output Kalman filter. Normal GPS location fixes are only exploited for the initialization of the particle filter. Map information is

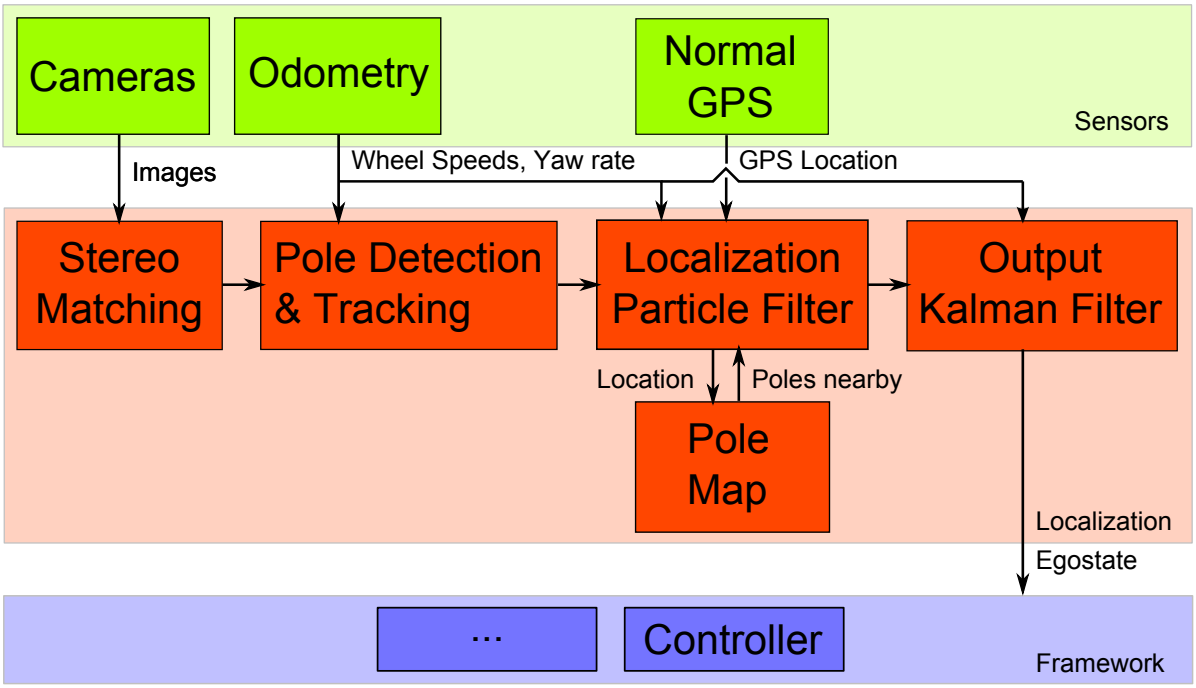
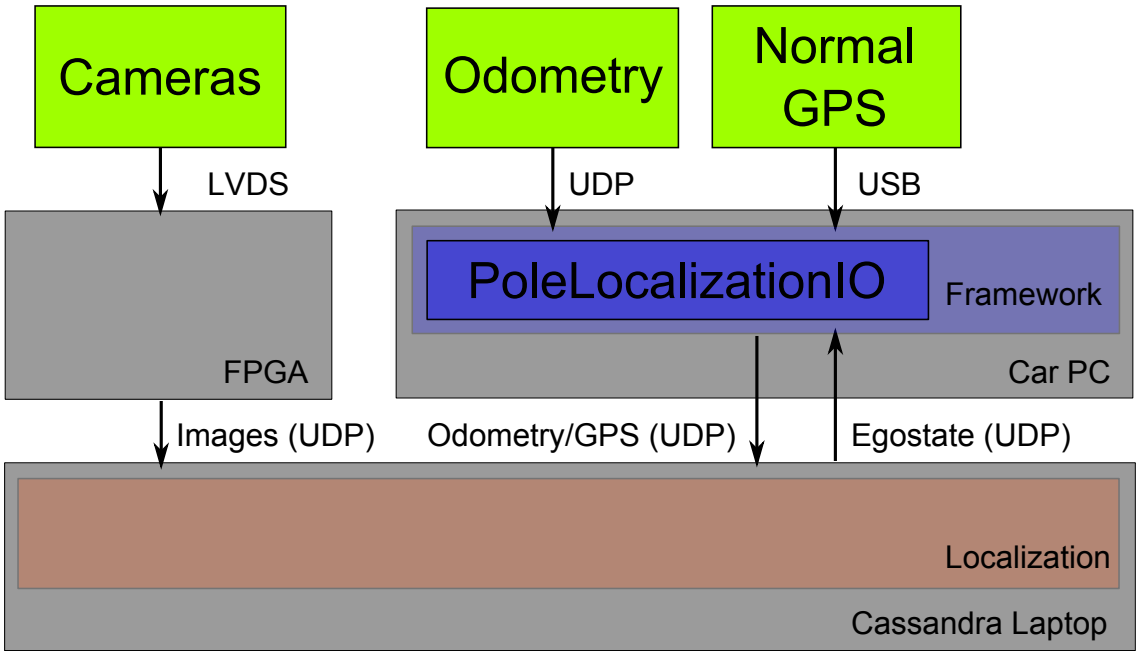


Figure 5.4: Localization dataflow with sensors, localization and framework and the main dataflows between those components

queried only by the particle filter, using its current best estimate. The output Kalman filter transmits its results to the autonomous car framework by means of an Egostate.

The components are distributed over several hardware entities (Fig. 5.5a). The localization algorithm is executed on a separate laptop, running in the Cassandra image processing framework. It receives its inputs from the FPGA and the Orocos laptop, which forwards odometry and GPS information. The Orocos laptop is used to run the rest of the tasks for autonomous driving, e.g. obstacle avoidance, vehicle controller or high level behavior. This separation is partly driven by resource constraints as the localization would take too many resources on the Orocos car PC for safe driving. Ease of implementation for the localization solution was another reason. Communication is mostly done by UDP sockets, providing a low-latency, low-overhead communication channel. They also offer an abstraction layer which can be used to add processing power to the test vehicle without changing the core processing nodes.

The experimental evaluation and field test results for this chapter are located in section 6.4 in the evaluation chapter.



(a) Deployment

Figure 5.5: Deployment of the components: The partition into three connected devices is partly driven by computing resource constraints. The transmission of sensor results by UDP makes them available at all processing nodes and simplifies reconfiguration.

6 Experimental Results

This chapter discusses the experimental results gathered to evaluate the chosen approaches. The first section presents the evaluation platform for autonomous vehicles. The second section is devoted to automotive stereo vision. Starting with online calibration, the analysis continues with wSGM and CS-CT. Rapid SGM closes the evaluation of the stereo processing. For localization and mapping, the same test areas are defined and the created maps are analyzed. Localization results are provided at the end of this chapter. A special section describes the accuracy measures for localization, as no off-the-shelf solution exists.

6.1 Evaluation Platform for Autonomous Driving

The test vehicle *Made in Germany* (MIG) is a Volkswagen Passat B6 equipped with a multitude of additional sensors and interfaces to the major actuators (Fig. 6.2, close-ups in Fig. 6.3). Nevertheless, the sensor integration approach had the goal of giving the vehicle a standard appearance (Figure 6.1) and use sensors that could be or are used in series production. The test platform was designed and built as part of the AutoNOMOS project ([118], p. 53), in order to evaluate the specific strengths and weaknesses of different sensor technologies available, start the development of new assistance systems and provide a testbed for autonomous driving.

Actuators necessary for driving are available through special gateways using the Controller Area Network (CAN) bus giving access to the state of accelerator and brake pedals, and the automatic gearbox. The steering is controlled through an additional motor attached to the steering column. Communication with other cars is available with headlamp control, indicator lights and the horn. The window lifters can be accessed as well. The gateways are two-directional, transmitting control commands and the current state of all actuators, as brake level, lighting state or steering wheel angle. A security concept includes an immediate hand-over, if driver intervention is detected.

Access to sensors part of electronic stability control is included as well. Wheel ticks

Figure 6.1: Test Vehicle *Made in Germany* (AUTONOMOS LABS Press Kit, http://autonomos-labs.com/media/press-kit/)

from all four wheels are available and the output of the yaw rate sensor that gives an angular velocity of the vertical axis of the vehicle. These two sensor types can be used for odometry calculation. The output of the four ultrasound sensors of the park assist system provides distance information for close ranges and low speeds. As a localization solution, the car includes an Applanix inertial navigation system coupled with a differential GPS, that uses an odometer for reliable speed estimates and real-time correction data.

Three different camera based sensors are available. A central front-facing camera, located at the central upper part of the windscreen, performs lane detection tasks, giving estimates of the position of the car in the ego lane and its course. This is a series sensor, only the state estimates are available. The detection range of the camera is up to 60 m, lateral deviation, heading angle, curvature and curvature rate are estimated. Next to it a stereo camera pair is installed. The monocular cameras with a baseline of 30 cm have a resolution of 768×480 pixels and provide a high dynamic range image of 12 bit intensity range with a frame rate of 22 Hz. Their field of view is approximately 50 degrees and they have a detection range of around 40 m, depending on the application. An additional pair of wide-angle color cameras is installed primarily for the purpose of traffic light detection. Their installation orientation is slightly rotated outwards to improve coverage for this purpose.

Two different types of rotating Lidar sensors are installed. A cluster of Ibeo Lux sensors gives tracked obstacle objects with distances up to 100 m and a viewing range of 110 degrees using 4 rays. They cover the front and back of the car completely, leaving only a small part at the side uncovered. A Velodyne sensor with 64 rays rotating at 15 Hz is used mainly as a reference system, as the resulting point cloud describes the near vehicle environment quite well. Its detection range is approximately 40 m.

Radar sensors complete the equipment of the test vehicle. Two different types are installed: A 24 GHz based short-range radar mainly used for blind spot applications in series production manufactured by Hella and two types of 77 GHz wide- range radars with a field of view of 12 degrees and a detection range of roughly 100 m. These are used typically for automated cruise control and are manufactured by TRW and SMS.

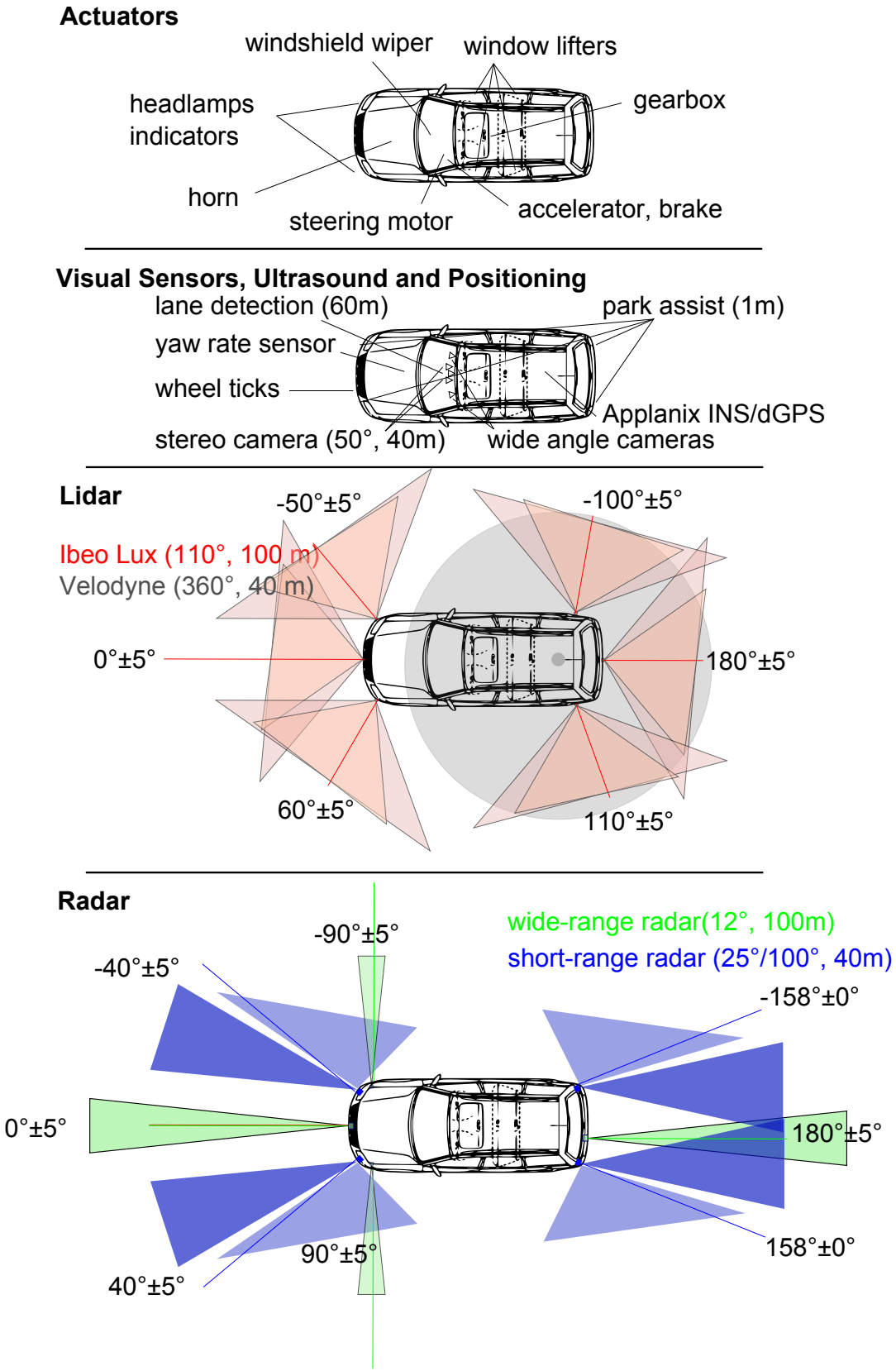
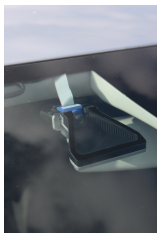


Figure 6.2: Actuators and sensors part of the test vehicle *Made in Germany* (MIG)

(a) Velodyne Lidar



(b) Inka Camera

(c) Guppy Camera

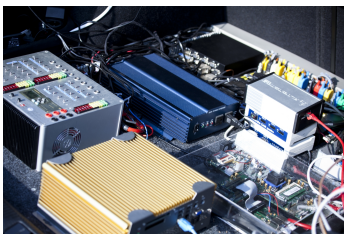
(d) Applanix GPS Antenna

(e) Hella Radar

(f) SMS Radar

(g) Ibeo Lux Lidar

(h) Camera Setup



(i) Trunk with ECUs, FPGA and computers

Figure 6.3: Sensor and computing devices close-ups and mounting

Wide-range radars are installed at the front and the back. The idea is to detect obstacles at far distances in the vehicle drive path and see obstacles at intersections as well. The short-range radars have two fixed beams and a range of 40 m. A more focused beam has a field of view of 25 degrees and the second one 100 degrees. The output of the short range radars are tracked object lists. Their purpose is to detect objects in the blind spot and enable the detection of vehicles approaching from behind on the ego-lane or neighboring lanes and to give a complete picture of the dynamic vehicle environment at intersections and in urban driving situations such as roundabouts.

The sensors and actors are all connected by an Ethernet network to the processing and control device, a 17" laptop workstation with four physical cores. It contains Raid-0 mirrored solid-state disks to record all incoming data and it uses an Ubuntu Linux as an operating system and Orocos as a process framework [112]. The person working on the laptop typically sits in the front passenger seat, monitoring the system.

The test vehicle is certified by the German technical control board (TÜV) and is approved for autonomous driving in the federal state of Berlin. This was achieved through the development and implementation of a safety concept guaranteeing a safe reaction of the car in emergency situations. In particular, actuator controls are monitored and are detached if errors are detected. A safe handover to manual control is possible at all times. A specially instructed test driver is necessary for each test drive. The car has driven autonomously on hundreds of test drives in environments ranging from German highways to small urban neighborhoods and has gained a reasonable amount of public attention, for example by driving autonomously from the campus of Freie Universität Berlin to the Brandenburg Gate.

6.2 Automotive Stereo Vision

6.2.1 Online Calibration

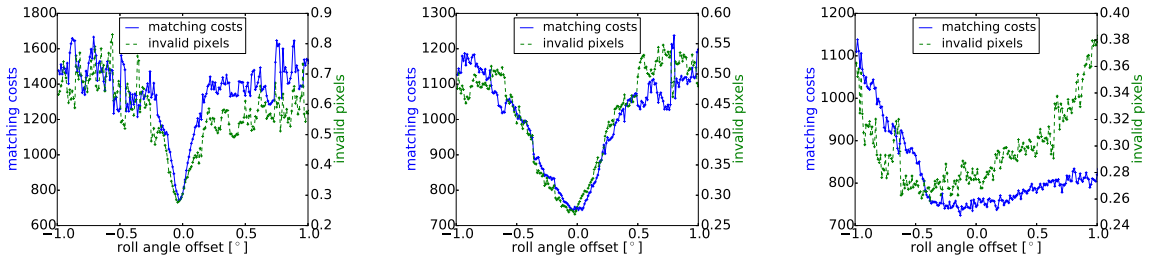
Major parts of this evaluation have been presented by myself in [106]. An understanding of the shape and smoothness of the target functions in the optimization space gives an idea as to which approaches should be pursued in order to find a minimum of the function. Fig. 6.4 depicts several slices of optimization space with variations of the three relative rotation angles. SGM costs are based on a 5×5 Census similarity criterion and fixed smoothing penalty parameters $P_1 = 7$ and $P_2 = 20$. As expected, the minima of the cost function are most pronounced for the pitch angle and a bit wider for the roll angle. The yaw angle only shows a minimum for SGM costs, but it is not very pronounced.

It can be observable that ELAS matching costs are a lot noisier than SGM costs. The difference in smoothness between ELAS and SGM might be caused by the higher robustness of SGM towards false matches and the inclusion of smoothness costs as part of the cost accumulation process. This leads to a more stable solution, whereas small changes in the input can lead to greater effects in ELAS. With SGM, the Monte-Carlo-Approach for optimization might not be necessary. Standard gradient descent would be an option as well. This could lead to speed ups in the calibration process, especially in connection with efficient implementations as rSGM.

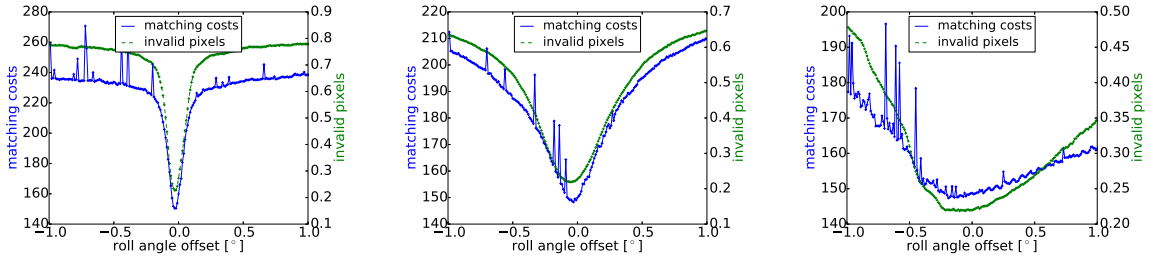
Varying two angles at once shows a rather smooth valley for pitch and roll, with no clear and continuous gradient direction being observed outside of this region (Fig. 6.4). The algorithm design for optimization has to take this into account. A pure gradient descent based approach might fail to find the global optimum, staying stuck in some local minimum in the surrounding region. The chosen variant of the Metropolis algorithm seems to be a good choice for both types of matching costs. It can furthermore adapt to the different sensitivity of the objective function to angle variation with e.g. larger step widths for the roll angle than for the pitch. If a standard gradient descent algorithm were used, as is possible for SGM, one would have to take care of this, possibly by including a momentum term.

A first validation of the algorithm was performed directly after an off-line calibration. It showed the expected zero offsets for pitch and roll angle. In order to test the performance and robustness of the algorithm, three other tests with real scenarios were performed. The first test scenario was recorded on the university campus and had a nearly optimal calibration. Scenario two had a rather strong roll de-calibration in the same environment. Scenario three covered bigger urban streets, inner city highways and tunnels. Calibration is assessed as good here. The algorithm was executed once per frame base with a spacing of roughly one second to mimic the expected usage.

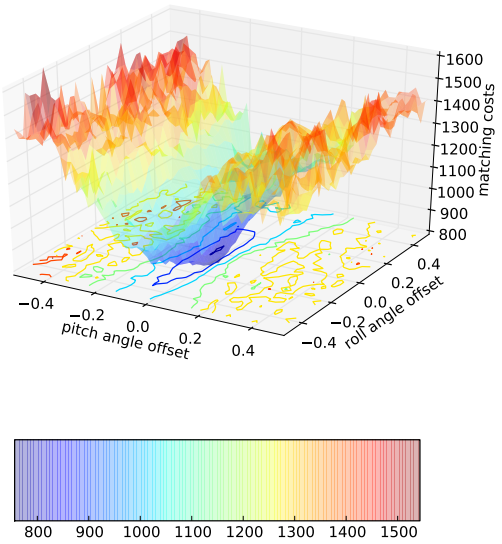
Since no ground truth on the correct calibrations is available for the test scenarios, the evaluation is based on the consistency of the estimation and visual inspection of the depth map. Consistency is evaluated based on the distribution of the estimated angles with standard deviations and extremes. The depth map should be dense and free of larger mismatched regions. Scenario two shows a good performance of the algorithm (Fig. 6.5). The standard deviation of the estimates is 0.005° for the pitch and 0.034°



(a) Pitch angle ELAS (b) Roll angle ELAS (c) Yaw angle ELAS



(d) Pitch angle SGM (e) Roll angle SGM (f) Yaw angle SGM



(g) Pitch and roll angle

Figure 6.4: Systematic modification of angles in the stereo rig by additional relative offsets with ELAS matching costs [35] (upper row) and SGM matching costs (middle row). Variation of roll and pitch in the lower row. Clear minima for pitch and roll angle exist, but pure gradient based methods are prone to fail, at least for ELAS costs. Yaw angle shows a very wide minimum.

Table 6.1: Optimal angle offset estimation for scenarios 1 to 3 - Results 1* and 3* are the results for scenario 1 and 3 with an additional valid frame filter, based on the valid pixel ratio.

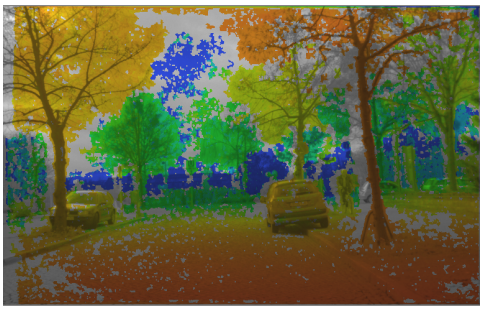
scenario	1	1*	2	3	3*
minimum pitch [°]	-0.114	-0.033	-0.105	-0.201	-0.036
maximum pitch [°]	-0.001	-0.014	-0.082	0.113	-0.009
mean pitch [°]	-0.027	-0.024	-0.095	-0.029	-0.026
stddev. pitch [°]	0.017	0.004	0.005	0.032	0.005
minimum roll [°]	-0.353	-0.160	-0.950	-0.456	-0.073
maximum roll [°]	0.043	0.043	-0.792	0.096	0.095
mean roll [°]	-0.059	-0.049	-0.866	-0.021	-0.001
stddev. roll [°]	0.065	0.042	0.034	0.075	0.030

for the roll angle (Table 6.1). As the angular resolution of the camera is approximately 0.06°/px, sub-pixel accuracy has been reached for both angles. The depth map has a higher density and parts with obvious false matches, as on the lower right side of the image, have been corrected. Scenario 1 exhibits a larger variation of the estimated values. Looking at the reason for this, one can identify frames with insufficient texture on the tarmac preventing a reliable matching of a larger part of the image, as in Fig. 6.6a and 6.6b. The road surface is usually rather hard to match because of the violation of the fronto-parallel assumption and the big change rate in disparity. The road surface is qualified as an indicator as to whether the calibration is good or not. If it cannot be matched at all, then there is a reduction in the differences of the matching costs for the tested calibrations. Hence, for good results, the road surface should possess enough texture to be matched.

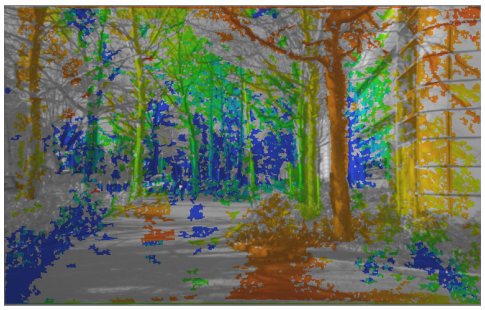
As the road surface covers a large part of the image, frames with insufficient texture can be invalidated using the fraction of valid pixels in the whole image as an indicator (Fig. 6.5c). If this fraction is below a threshold, the online calibration result of this frame is not used for the integrated estimate. Another option would be a measurement of texture, like the strength of the structure tensor [12], but this would have to be adapted to the employed matching method, as the necessary texture for a successful matching varies from algorithm to algorithm.

With valid pixel filtering, sub-pixel accuracy is reached in scenario 1 as well. The density of the depth map is increased slightly, e.g. at the cobblestone parts of the street. In Scenario 3 comparable problems occur, that are fixed by the same solution. The improved depth map of Scenario 3 shows no obvious large false matches. Tunnel scenes as in Fig. 6.6c allow no matching of a big fraction of the frame due to low lighting, missing texture and strong motion blur influence, and are filtered out. Some frames are incorrectly treated as not valid by the method, as in Fig. 6.6d.

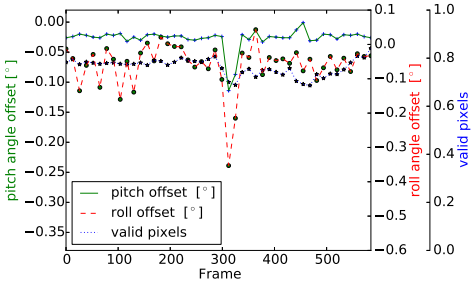
Using a simple mean filter for the integration of the per-frame results seems justified. The errors are comparably low. If the scenarios with a higher percentage of low texture



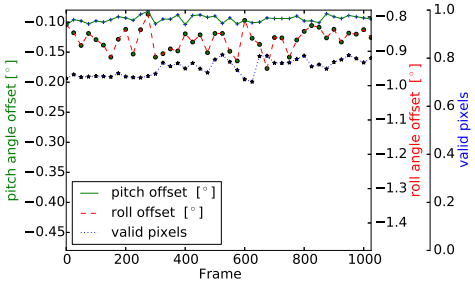
(a) Scenario 1 - initial depth map



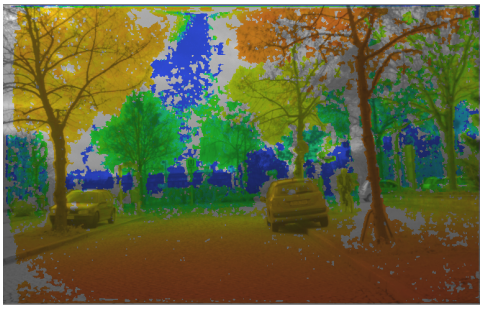
(b) Scenario 2 - initial depth map



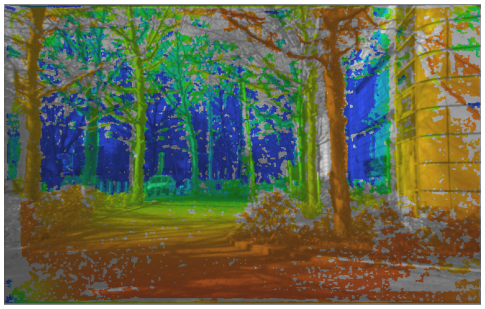
(c) Scenario 1 - weak de-calibration



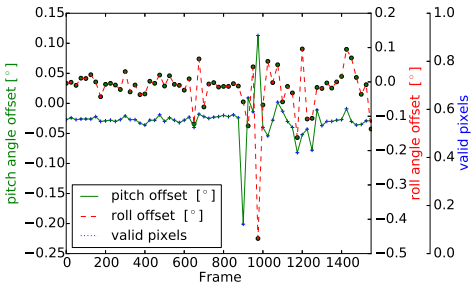
(d) Scenario 2 - strong roll de-calibration



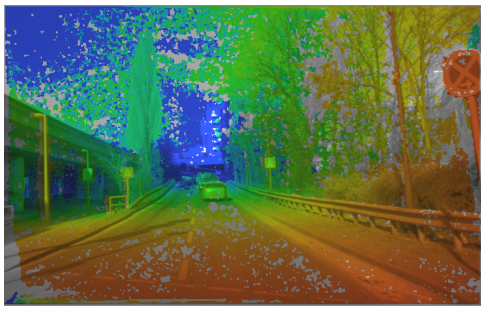
(e) Scenario 1 - improved depth map



(f) Scenario 2 - improved depth map

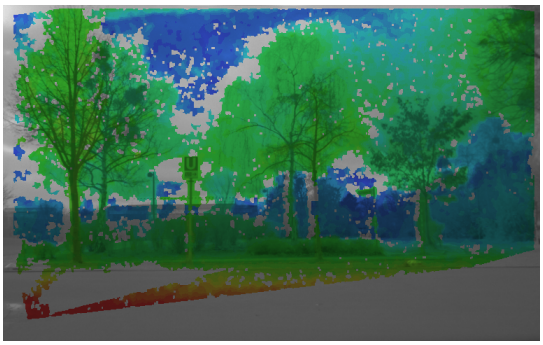


(g) Scenario 3



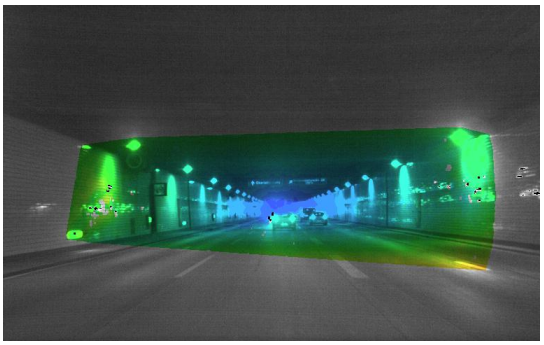
(h) Scenario 3 - improved depth map

Figure 6.5: Roll and pitch angle offset estimation for different scenarios: Using an ELAS-based matching cost measure, the improved depth map is created using mean estimated offsets for pitch and roll angle. Depth maps are color-coded overlays on the base images.

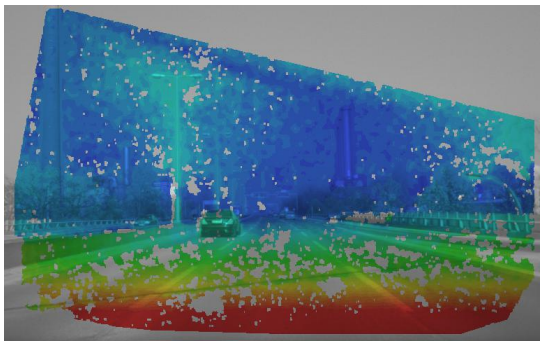


(a) Scenario 1 - frame 312

(b) Scenario 1 - frame 455



(c) Scenario 3 - frame 975



(d) Scenario 3 - frame 1525

Figure 6.6: Example failure frames of the online calibration: Due to lack of contrast, no depth information could be computed for a large part of the images. Therefore, they are invalidated as unreliable.

surfaces prevail, the availability could start to be problematic. Dynamic thresholding of matching costs is an option to cope with this or a Kalman filter with gating. In rural areas, the upper part of the image can comprise of a lot of sky. This should be excluded as it often does not exhibit enough structure to be reliably matched. However, the overall system availability in urban scenarios is sufficient even with the simple approach.

6.2.2 Weighted SGM and Center-Symmetric Census Transform

Major parts of this evaluation have been presented by myself in [107]. The results on the KITTI training set are used as a reference measure. As a baseline SGM algorithm, my own implementation was used with Census as matching cost and a window size of 9×7 pixels (SGM $CT_{9,7}$). It sums 16 paths and employs the left-right check with diagonal search. Depth discontinuity is regularized with a linear penalty function for P_2, which depends on the image intensity differences along the path like in [6]. A gravitational constraint [31] is applied to disambiguate areas like sky or road surfaces and increase consistency in vertical regions. The parameters setting is $P_1 = 7, P_{2_{min}} = 17, \alpha = 0.5, \gamma = 100, P_G = 3$. Results are ahead of the OpenCV SGM variant (Table 6.2, parameters equal to KITTI website), which can be credited to the inferior SAD matching

Table 6.2: Comparing different SGM approaches with the KITTI training data set: the baseline algorithm, variations of the Census transform and weighted SGM (Out-Noc: outliers non-occluded pixels, Out-All: outliers all pixels)

	2px		**3px**		
Method	**Out-Noc**	**Out-All**	**Out-Noc**	**Out-All**	**Density**
OpenCV SGM	11.40 %	12.92 %	8.39 %	9.81 %	85.50 %
SGM $CT_{5,5}$	10.90 %	12.28 %	7.34 %	8.54 %	88.74 %
SGM $CT_{9,7}$	9.39 %	10.80 %	6.23 %	7.44 %	91.53 %
SGM $Sparse\text{-}CT_{9,7}$	9.70 %	11.15 %	6.61 %	7.87 %	91.49 %
SGM $CS\text{-}CT_{9,7}$	9.59 %	11.06 %	6.51 %	7.76 %	91.97 %
SGM $WCS\text{-}CT_{9,7}$	9.12 %	10.47 %	6.03 %	7.17 %	92.12 %
SGM $HWCS\text{-}CT_{9,7}$	9.09 %	10.44 %	6.05 %	7.20 %	92.18 %
wSGM $WCS\text{-}CT_{9,7}$	8.99 %	10.35 %	5.90 %	7.04 %	91.99 %
wSGM $HWCS\text{-}CT_{9,7}$	8.89 %	10.25 %	5.89 %	7.04 %	92.17 %

costs used in OpenCV. To evaluate the quality increase of the two sparse Census variants, a 5×5 Census transform was implemented as well (SGM $CT_{5,5}$, adapted parameters, tuned to be optimal). Both sparse encodings show a rise in matching quality, CS-CT being a bit better than Sparse-CT.

For the weighted CS-CT, two variants were tested, one with only additional horizontal weights (HWCS) and one with full weights (WCS), with each weighted region being 3 pixels wide. Both approaches lead to better results than the normal Census transform, resulting in a higher matching density with less outliers.

Weighted SGM was combined with these matching costs as well and this results in additional improvements. A weight factor of 3 was used for preferred paths and one for the other paths. Only the results using the generative probabilistic model are reported. The sub-sampled SGM version did not lead to any improvements. Restricting the weight adaption to the horizontal and vertical paths resulted in a slightly better depth estimation. In urban scenarios, the other surface orientations are not that often, and large and much more difficult to estimate reliably with a sub-sampling approach due to their limited size. Examining the differences in detail (Fig. 6.7), one can deduce that the weighted Census transform diminishes the probability of outliers overall, whereas Weighted SGM leads to a reduction of larger areas with false matches at certain, single frames (see Figure 3.6). The poor contrast in the shown frame results in a larger area with mis-propagated disparities on the tarmac with SGM. wSGM raises the weights for the horizontal paths and can construct the real surface.

For the KITTI test set, wSGM combined with WCS (Table 6.3, with additional interpolation) exhibits a comparable performance to iSGM and a marked improvement on the SGM baseline algorithm. Its execution is comparable or better to the closest competing approaches using a C++ implementation and no specific SIMD processor instructions or multi-threading. The algorithm clearly shows progress towards robustness

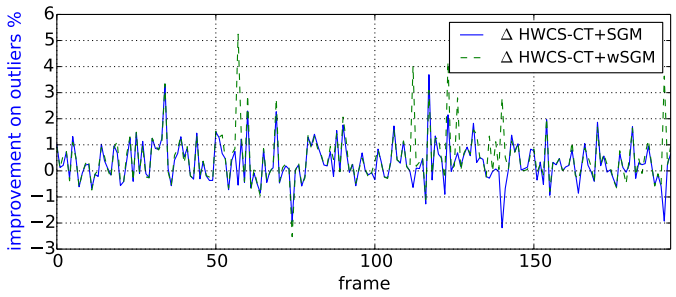
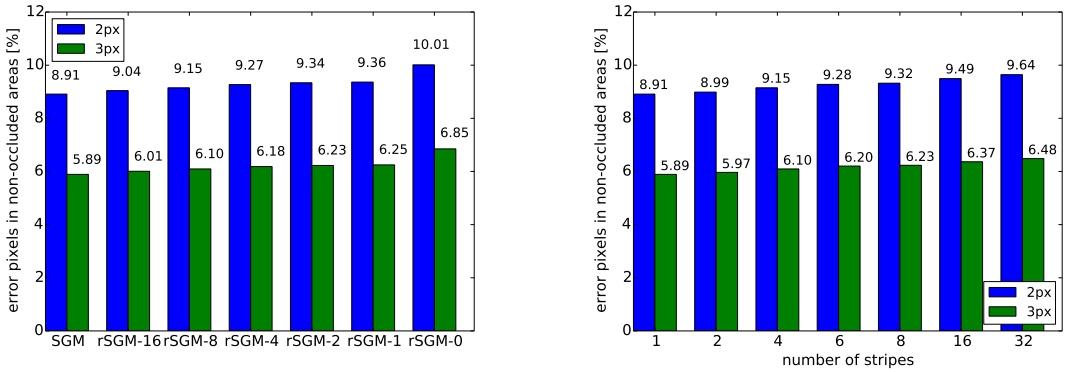


Figure 6.7: wSGM improvement per frame to SGM $CT_{9,7}$ on the KITTI training set (% outlier pixels $> 2px$)

in the family of SGM based algorithms. The propagation of disparities is driven by the three-dimensional shape of the surroundings in less textured areas, leading to a smaller probability of false matches.

6.2.3 Rapid SGM

Major parts of this evaluation have been presented by myself in [108]. At first, I tested the quality degradation induced by striping. Looking at the quantitative side with the KITTI training data, a relationship between border width and matching quality is perceptible. The quality increases as the border width gets higher. Using a width of 16 px the deviation to the standard SGM version is hardly noticeable. Without an additional border area, the fraction of wrong pixels rises by more than 1% (Fig. 6.8a) compared to the non-striped SGM method. A border width of 16 pixels diminishes this increase in the error to 0.12% above the 2 px error threshold and 0.13% above the 3 px error threshold.



(a) 4-striped SGM on the KITTI training data set - relationship between border width and quality

(b) Striped SGM with a border of 8 px on the KITTI training data set - variation of the number of stripes

Figure 6.8: Influence of border widths and number of stripes

Table 6.3: Evaluation on KITTI test set with error threshold 3 px. The table shows the top 17 ranking methods as of middle of January 2014 (submission date of first publication) and the OpenCV Implementation of SGM as rank 27. SGM-based methods are highlighted in bold: *Weighted Semi-Global Matching* wSGM [107], our *rapid SGM* with 4 stripes and a border of 16 rSGM, *Iterative Semi-Global Matching* iSGM [44], OCV-SGBM2 (anonymous submission), *Semi-Global Matching* SGM [45] and *OpenCV Semi-Global Block Matching* OCV-SGBM.

Rank	Method	Out-Noc	Out-All	Avg-Noc	Avg-All	Runtime	Environment
1	SceneFlow	2.98 %	3.86 %	0.8 px	1.0 px	6 min	4 cores @ 3.0 Ghz (Matlab + C/C++)
2	PCBP-SS	3.40 %	4.72 %	0.8 px	1.0 px	5 min	4 cores @ 2.5 Ghz (Matlab + C/C++)
3	gtRF-SS	3.83 %	4.59 %	0.9 px	1.0 px	1 min	1 core @ 2.5 Ghz (Matlab + C/C++)
4	StereoSLIC	3.92 %	5.11 %	0.9 px	1.0 px	2.3 s	1 core @ 3.0 Ghz (C/C++)
5	PR-Sf+E	4.02 %	4.87 %	0.9 px	1.0 px	200 s	4 cores @ 3.0 Ghz (Matlab + C/C++)
6	PCBP	4.04 %	5.37 %	0.9 px	1.1 px	5 min	4 cores @ 2.5 Ghz (Matlab + C/C++)
7	PR-Sceneflow	4.36 %	5.22 %	0.9 px	1.1 px	150 sec	4 core @ 3.0 Ghz (Matlab - C/C++)
8	**wSGM**	4.97 %	6.18 %	1.3 px	1.6 px	6s	1 core @ 3.5 Ghz (C/C++)
9	ATGV	5.02 %	6.88 %	1.0 px	1.6 px	6 min	>8 cores @ 3.0 Ghz (Matlab + C/C++)
10	**rSGM**	5.03 %	6.60 %	1.1 px	1.5 px	0.3 s	4 cores @ 2.6 Ghz (C/C++)
11	**iSGM**	5.11 %	7.15 %	1.2 px	2.1 px	8 s	2 cores @ 2.5 Ghz (C/C++)
12	AARBM	5.14 %	6.20 %	1.1 px	1.2 px	0.4 s	1 core @ 3.0 Ghz (C/C++)
13	ALTGV	5.36 %	6.49 %	1.1 px	1.2 px	20 s	GPU @ 2.5 Ghz (C/C++)
14	**OCV-SGBM2**	5.38 %	6.50 %	1.0 px	1.2 px	2 s	1 core @ 2.5 Ghz (C/C++)
15	AABM	5.42 %	6.52 %	1.1 px	1.3 px	0.43 s	1 core @ 3.0 Ghz (C/C++)
16	RBM	5.50 %	6.48 %	1.2 px	1.3 px	0.2 s	1 core @ 3.0 Ghz (C/C++)
17	**SGM**	5.76 %	7.00 %	1.2 px	1.3 px	3.7 s	1 core @ 3.0 Ghz (C/C++)
27	**OCV-SGBM**	7.64 %	9.13 %	1.8 px	2.0 px	1.1 s	1 core @ 2.5 Ghz (C/C++)

If the number of stripes is increased and border kept constant, the results shows the expected quality degradation (Fig. 6.8b). As the covered image parts get smaller, the probability of separating a less textured area with a border between two stripes increases. However, the increase is only 0.59% above the 3 px error threshold and 0.73% above the 2 px error threshold when comparing full SGM to a 32-striped SGM. This shows that the approach is scalable to more than just a few stripes. Admittedly, at 32 stripes each stripe has a height of only 12 px, which means quite a huge overhead is brought about by the additional borders. If the image dimensions are a lot wider than they are high, vertical striping instead of horizontal could be considered.

Visual inspection gives a hint as to the regions where quality degradation appears. If only a small additional overlap is granted, the borders between the image stripes are visible in several regions: on the street, the pavement and in the treetops (Fig. 6.9, on the left, extract on the right). Less textured areas especially are prone to catch the disparity values of their more textured surroundings. If the stripes disconnect less textured areas from more textured ones, discontinuities in the disparities are obvious. If the overlap area is increased, the disparity discontinuities become smaller and fade. However, in scenarios containing very large untextured regions, these might prevail, as information is missing from the neighboring stripe. In this case, the normal SGM does result in a propagation of the disparities in the full image neighborhood sampled by the paths. This propagation is dominated by the regularization penalties, resulting in possible streaking artifacts starting from the most textured area. Depending on the depth and texture distribution in the scene and the regularization penalties, each one of them can better represent the true depth values. If a robust depth estimation is the goal, larger untextured areas should be invalidated to limit the influence of matching errors.

The influence of disparity compression is illustrated in Fig. 6.10. Depth values on slanted surfaces like on the barrier on the right side are a bit noisier with some visible steps. A sub-sampling of four leads to a reduced percentage of matched pixels. The quality degradation with factor two is not that apparent. Moreover, the strength of the left-right-check is diminished. The pole in the scene is less distinct and becomes a bit wider in comparison to the version that is not disparity compressed. This appears to be a result of indirect image sub-sampling in the disparity domain of the matching cost during the computation of the modified cost cube. On the training data set of Kitti, the quality is reduced by 0.33% for a sub-sampling of 2 and by 1.34% for a sub-sampling of 4 for the 2 px error threshold (Table 6.4). Especially the first result is striking, as the method does not aim for a pixel-accurate solution in areas near the ego vehicle.

Speed Evaluation

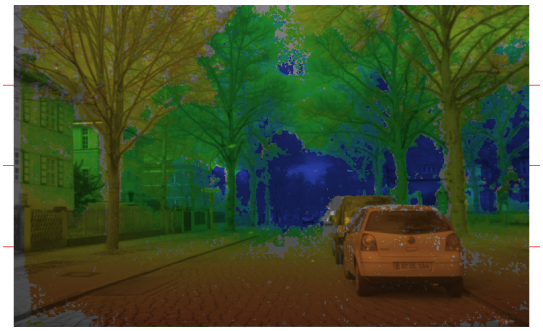
The speed evaluation is split into several parts.Firstly, the influence of the input image intensity depth is estimated. Secondly, the full SGM solution is compared to other implementations. The improvements by striping and disparity compression follow. All tests were done on a mobile Intel i7-4960HQ CPU with a base frequency of 2.6 GHz.

The input images used for the speed analysis have an intensity resolution of 12 bit.

(a) pavement detail - border 1 px

(b) border 1 px

(c) pavement detail - border 32 px

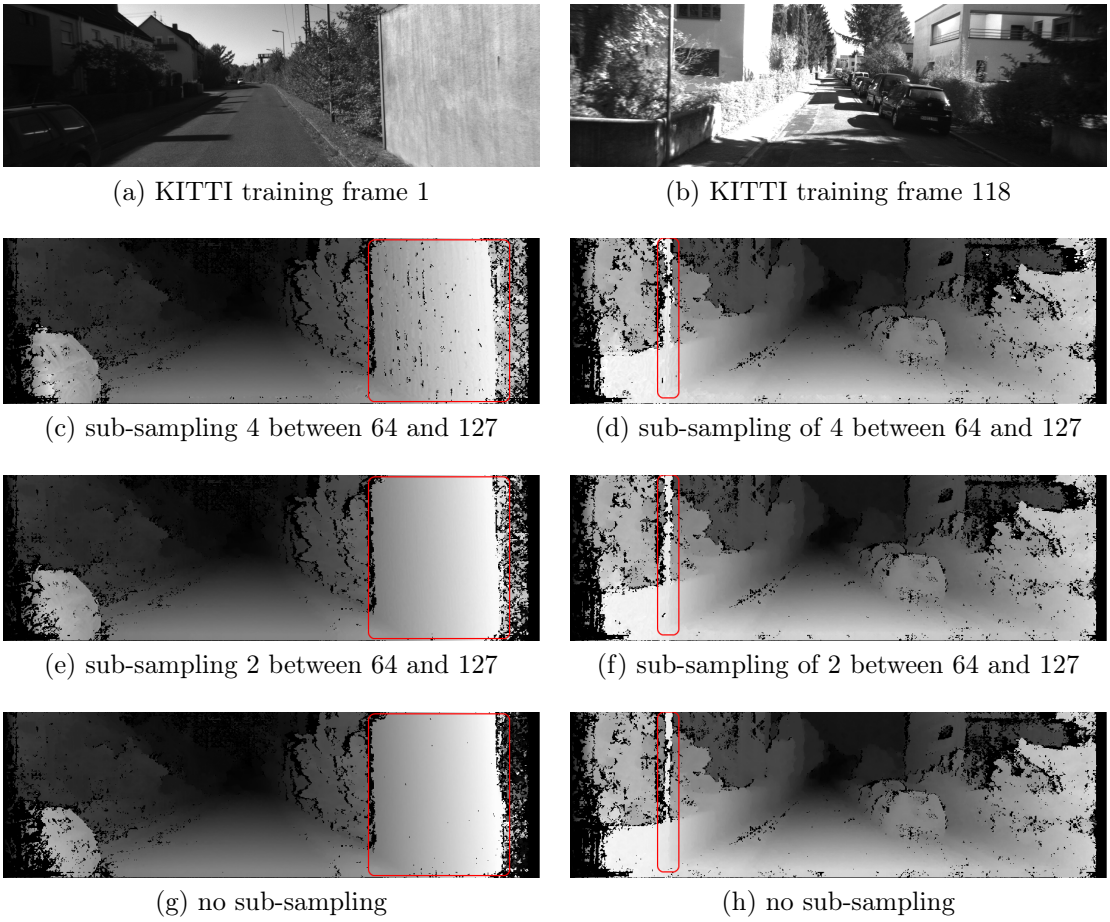


(d) border 32 px

(e) pavement detail - full SGM

(f) full SGM

Figure 6.9: Results of 4-striped SGM: full images and detailed view of pavement area with striping effects



(a) KITTI training frame 1

(b) KITTI training frame 118

(c) sub-sampling 4 between 64 and 127

(d) sub-sampling of 4 between 64 and 127

(e) sub-sampling 2 between 64 and 127

(f) sub-sampling of 2 between 64 and 127

(g) no sub-sampling

(h) no sub-sampling

Figure 6.10: Effects of disparity compression: In the images on the left the concrete barrier on the right side possesses disparities from 60 to 127. The pole in the images on the right shows less clear, smooth borders with bigger sub-sampling. Interesting areas are enclosed in rounded rectangles in the disparity images.

Table 6.4: Influence of disparity compression tested on the KITTI training data set: Out-Noc is the percentage of non-occluded pixels with a disparity error bigger than the threshold. SGM-DC2 uses a disparity compression of 2 for the upper 64 disparity values. SGM-DC4 uses a disparity compression of 4 for this range.

Method	Out-Noc 2 px	Out-Noc 3 px
SGM	8.91%	5.89%
SGM-DC2	9.24%	6.09%
SGM-DC4	10.25%	6.71%

Eight bit images can be processed as well. The intensity resolution does not have an apparent effect on the algorithm execution times, as only the Census transform and the image access for the P_2 penalty is affected. The differences in execution time stay below 1 ms.

The execution times for the standard SGM formulation show that my solution (denoted as SGM in Table 6.5) clearly surpasses the CPU implementation in [33], although only two as opposed to four cores are utilized. This is not only an effect of the newer and more powerful processor platform, but of the way parallelism is used in the implementation concept. Synchronization is limited to a coarse level and therefore the cores can run freely without waiting for each other. The solution is quicker than the early GPU implementation of [29] and a bit slower than [75], but consumes maximally only 47 W. For larger disparity ranges, the solution nearly achieves the speed of the FPGA implementation described in [32], if the metric is the number of calculated disparities per second.

Looking at the separate timings of the algorithm parts in Table 6.6, an almost direct linear dependency of the data cost calculation and disparity selection from the number of disparities to calculate is observable. The WTA for the base (left) image is faster than the one for the right image, as an additional data alignment step is needed. If the right disparity image is not necessary and an increase in quality can be spared by using the left-right check, then the runtime can be significantly reduced for WTA. The path accumulation step is also linear dependent with a factor slightly below one and it consumes the largest amount of the execution time. The post-filtering steps and the sub-pixel interpolation have a very limited influence on runtime. Even median filtering can be performed really fast on modern processor architectures.

The solution utilizing image striping with additional borders is able to achieve shorter frame cycle times, as all 4 available physical cores can be used throughout the complete algorithm. This method is called rapid SGM (rSGM). A frame rate of 19.6 Hz is reached for VGA resolution and 64 assessed disparities, and of 11.9 Hz for 128 disparity values. Using disparity compression, a processing speed enabling 16 Hz is possible with 128 disparities calculated. This type of depth sub-sampling produces little overhead. As one can see the number of evaluated disparities stays high and therefore it provides a significant increase in speed in comparison with the normal SGM version.

Table 6.5: Matching algorithm timings: rSGM uses 4 stripes and a border of 16 pixels. rSGM-DC2 additionally uses a disparity compression of 2 for the upper 64 values. rSGM-DC4 uses a disparity compression of 4 for the upper 64. values

Method **Unit**	**Cores**	**image size [px]**	**disparities [px]**	**cycle [ms]**	**disp/s [$10^6/s$]**
GPU [29]		320×240	64	76	64
GPU [75]		640×480	64	85	230
FPGA [32]		2·340×200	$2 \cdot 64$	40	218
FPGA [22]		640×480	64	30	648
CPU [33]	4	640×320	128	224	117
SGM	2	640×480	64	105	187
SGM	2	640×480	128	184	214
rSGM	4	640×480	64	51	390
rSGM	4	640×480	128	84	468
CPU [33]	4	268800	128	69	65
rSGM-DC2	4	640×480	128	71	415
rSGM-DC4	4	640×480	128	62	396

Table 6.6: Detailed timings for the full SGM variant

Algorithmic step **#disparities** **time**	**640×480** **64** **[ms]**	**640×480** **128** **[ms]**	**1248×384** **128** **[ms]**
5×5 Census	1	1	1
Data cost calculation	7	14	21
Path accumulation	76	137	217
WTA left & right	16	30	46
Median filtering	1	1	2
Left-right consistency check	1	1	2
Sub-pixel interpolation	2	2	5

6.3 Landmark Detection and Mapping

6.3.1 Evaluation Areas

In order to study the performance of the localization in typical urban environments, six test routes in five test areas in and near Berlin were chosen (see Fig. 6.11 and Table 6.7). A small suburban avenue with two lanes and weak markings in Kleinmachnow with detached houses is the first test site. Two round courses on the university campus in Berlin-Dahlem, paved mainly with cobblestones and without lane markings serve as the next two routes (called Campus Short and Campus Long). The Englerallee, a larger urban avenue with lane markers is the next area. To include also larger, three lane urban streets, the Straße des 17. Juni near the Brandenburg Gate was included as well. A small round course in the governmental area of Berlin with lots of pedestrians and bicycles completes the selection.

Table 6.7: Evaluation area properties

Name	**Length of Route(s)**	**Type**	**Type of Pavement**	**Markings, #Lanes**
Kleinmachnow	2.9 km	suburban neighborhood	tarmac	weak,2
FU Campus	1.1 km/1.4 km	small avenue	cobblestones & tarmac	none, 1
Englerallee	1.0 km	bigger avenue	tarmac	moderate, 2
Straße des 17. Juni	3.4 km	major street	tarmac	good, 6
Reichstag	1.3 km	small street with pedestrians, bicycles	tarmac	good, 2-6

The selection includes therefore a wide variety of different environment settings that can be encountered in a European city. The road class and the pavement is varied. The number of available pole-like structures and their geometrical distribution in relation to the vehicle therefore also varies. Objects with a pole or tree-like shape that can be mistaken as landmarks like garbage bins, bicycles or pedestrians vary in their observation probability as well. Traffic density is another parameter that is a lot lower in the smaller streets than on the larger ones, making the detection of poles a lot harder due to occlusions. Different types of curves are included as well, from very slight ones in Kleinmachnow, to mainly ninety degree corners on the campus to u-turns on Englerallee.

6.3.2 Mapping Results

Fig. 6.12 shows maps of the test areas with an overlay of the landmarks. The areas vary significantly in their distribution of landmarks. While Kleinmachnow and Englerallee show a quite uniformly distributed density, Reichstag and Straße des 17. Juni fluctuate

(a) Kleinmachnow

(b) FU Campus

(c) Englerallee

(d) Straße des 17. Juni

(e) Reichstag

Figure 6.11: Evaluation areas: example frames from the left camera of the stereo rig

to a greater extent in landmark density and have a lower number of pole-like structures per length of the route. The number of obvious clutter poles is very low. The clusters of poles that appear now and again are often traffic signs, bollards or traffic light posts that are mixed in amongst normal trees. Even in rather regularly patterned trees like on the campus, from time to time irregularities show up, making the disambiguation of the position during localization easier. The approach does not only rely on the presence of trees to gather the necessary amount of landmarks, but the other types of posts contribute as well.

In order to get some descriptive statistics on the maps I define the following measures:

Table 6.8: Map statistics for the test areas

Name	**Pole Density** (Mean) [1/m]	**Sightings/Pole** (Median)	**Matchable poles** (Mean)
Kleinmachnow	0.21	21	6.66
FU Campus	0.28	28	17.71
Englerallee	0.15	34	6.39
Straße des 17. Juni	0.08	19	2.02
Reichstag	0.24	16	4.14

pole density, number of views per mapped pole and the number of matchable poles. The pole density is simply the number of mapped poles per distance driven during mapping. The number of sightings for each pole relates to the number of times the pole was seen during tracking. We derive the number of matchable poles from driving along hypothetical bins on the mapping trajectory. Each bin is hereby 1 m long. For each bin, we query the poles within a 30 m reach from the map and in the viewing area of the stereo rig. With this measure we can get an idea of how many poles can be matched during a localization run in this map.

Table 6.8 shows the results for the test areas. While Kleinmachnow and the campus area have statistics that indicate an easier setting for localization, the areas in the city center show lower numbers of pole density and especially matchable poles. The Straße des 17. Juni has an extremely low number with approximately 2 matchable poles, due to the high lateral distances to the observed trees.

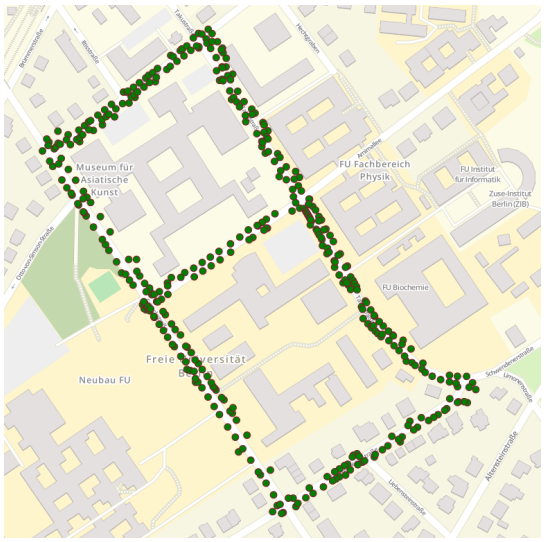
6.4 Localization for Autonomous Vehicles

As the localization solution seeks to run as fast as possible but relies on high-quality disparity maps as well, the choice of the stereo matching algorithm should be clarified. We use rSGM with 4 stripes, a border width of 16 pixels and 5×5 Census as matching cost. This fits well with our angular resolution and leads to the best results at comparably low processing times. The matching parameters are $P_1 = 7, P_{2_{min}} = 17, \alpha = 0.25, \gamma = 50$. The default parameters for the particle filter are $M = 1000, \alpha_1 = 0.05\,m/s, \alpha_2 = 1.06\,^\circ/s, \alpha_3 = 0.1, \alpha_4 = 1\,^\circ/s$, $\beta_p = 1/60, \sigma_w = 0.1\,m, \kappa(z^j) = 1, p_D = 0.8, T_{N_{eff}} = 500$ and for the output Kalman filter $\sigma_{o,v} = 0.1\,m/s, \sigma_{o,\dot{\psi}} = 0.3\,^\circ/s, \gamma_p = 1$.

6.4.1 How to Measure Localization Accuracy?

Using a DGPS Reference System

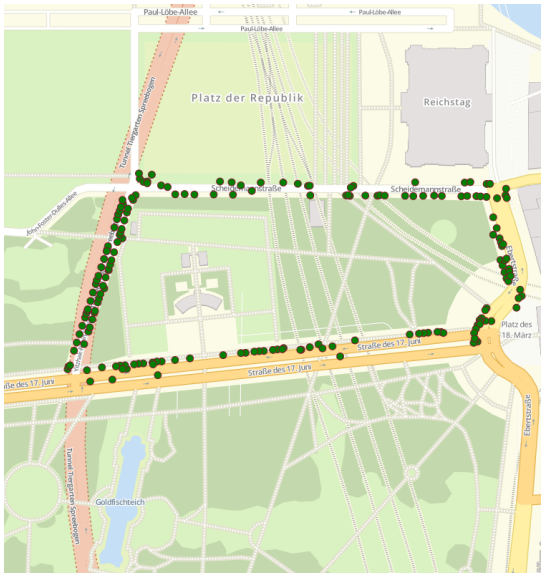
The available reference system is an *Applanix POS LV 510*. This is a differential GPS (DGPS) system with an inertial measurement unit (IMU) and real-time kinematic (RTK) data provided by a UMTS connection [90]. Odometry information is obtained by an ad-



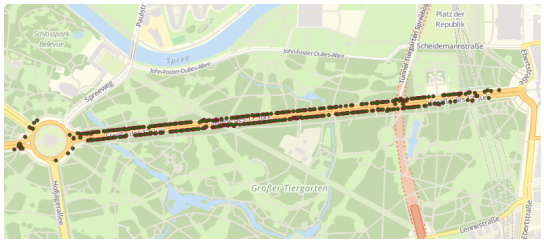
(a) Campus - the short route only covers the upper rectangle, the long route is the enclosing polygon.

(b) Kleinmachnow

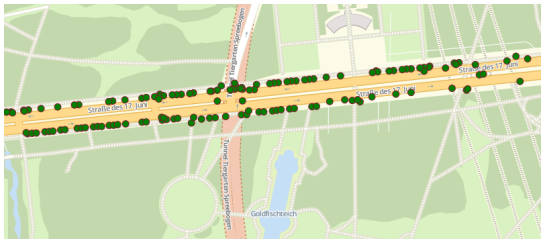
(c) Englerallee



(d) Reichstag



(e) Straße des 17. Juni



(f) Straße des 17. Juni (close-up)

Figure 6.12: Maps of the test areas: Landmarks are displayed as green circles on top of an Open Street Map base layer (© OpenStreetMap contributors, www.openstreetmap.org/copyright)

ditional wheel encoder. Position and velocity estimations are available with a frequency of 100 Hz.

What level of accuracy can be expected from such a system? Several test runs in urban environments were conducted. Fig. 6.13a plots the result of a shorter test run. The average positional uncertainty, measured by the size of the error ellipsoid, is 0.13 m for the small side of the ellipse and 0.26 m for the big one. The maximum values are 0.22 m and 0.35 m. The ellipsoid describes an uncertainty region of one standard deviation [91]. Accuracies below 0.1 m are usually only reached during vehicle standstill. Figure 6.13b plots a longer route showing similar properties. It can be seen that those accuracies can only be reached, if ten or more satellites are available. Longer periods with a lesser number have a negative influence on the reported accuracies. However, in typical urban scenarios with limited satellite visibility and multi-path effects, the uncertainty figures of the system can be misleading. Drift of the navigation solution is possible (Fig. 6.13c) in the dimension of several meters despite reported localization accuracies being clearly below the strength of the drift. Small position jumps in the range of 0.2 m happen from time to time and sudden changes between the different quality levels of the real-time kinematics (float and fixed RTK), mainly at standstill, also lead to jumps in the position information.

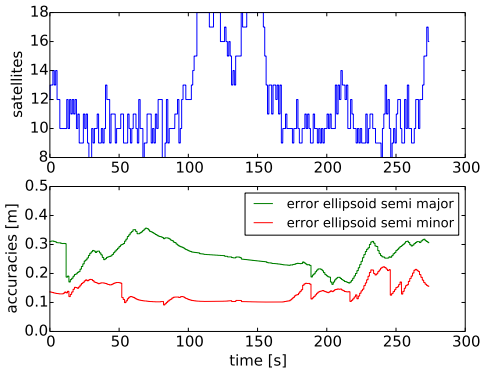
Therefore, the usefulness of the DGPS for accuracy estimates is limited. As the measurement system is not several times more accurate than the localization solution to be evaluated and has a high probability of outliers with even worse accuracies, the position information cannot be used as a reliable reference. The speed and yaw rate estimates can be used, as they are mainly based on the odometry sensors that are part of the INS. In order to compare the results of the position estimation and give an indication on the accuracy levels that can be expected, two other measures will be described in the next subsections: repeatability and distance to lane markers in the map.

Repeatability Measure

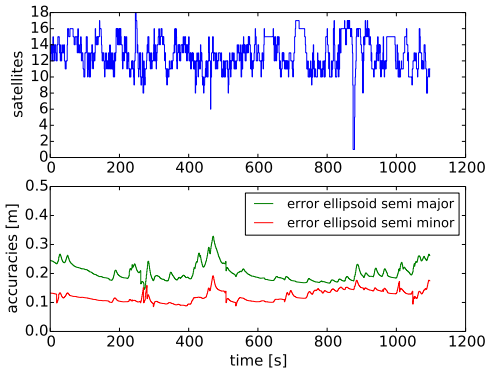
The concept behind a repeatability measure is that the localization algorithm shall come to the same result if the vehicle is at the same position. So, if the car drives on exactly the same route for several laps, then the trajectories of the laps should ideally be the same and one can measure the lateral error as the standard deviation of the minimal lateral distances of all laps. To remove all influences of automatic lateral control, test runs were performed with a human driver.

In practice, the driver cannot drive completely accurately and sometimes has to adapt the course of the vehicle due to other traffic participants. The typical human driver has a standard deviation of the lateral position while driving of 0.2 m with a range of 0.10 m - 0.27 m [122]. As the experiments were performed by a driver told to aim for repeatability and the test runs were comparably short, the resulting error might be lower, but errors of at least 0.1 m produced by the testing procedure seem probable. Areas with larger deviations, e.g. due to obvious evasive maneuvers, have to be excluded during the analysis and calculation of the aggregated values.

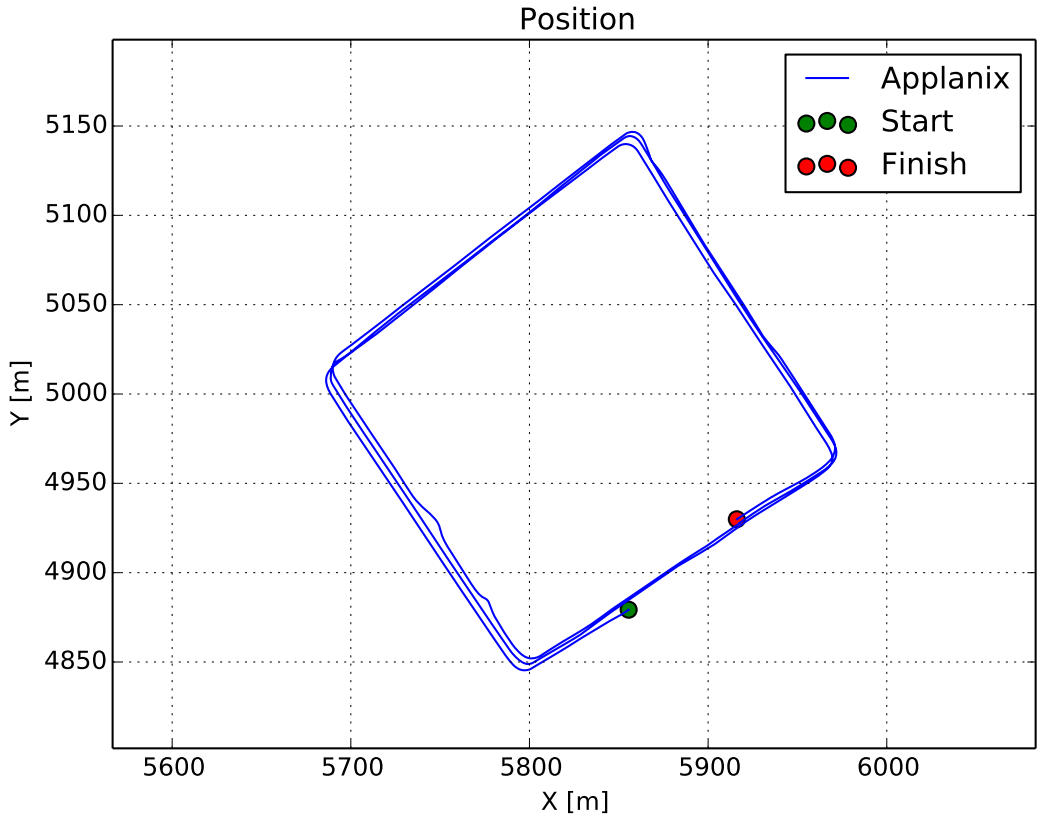
As the measure does not depend on a reference, only on the availability of several laps



(a)



(b)



(c)

Figure 6.13: Accuracy estimates of the Applanix DGPS with RTK in connection with the number of visible satellites in urban environments for test drives in Kleinmachnow (a), from Kleinmachnow to the university campus (b). An example of the Applanix position drift is given in (c). Several laps were driven on the test route Campus Short. The position of the Applanix drifts several meters between the first and the last lap in (c).

Table 6.9: Repeatability measure results - Values for Particle Filter and Kalman Filter are averages of 50 Monte Carlo runs.

Name	Particle Filter	Kalman Filter	Applanix	Laps
FU Campus Short	0.177 m	0.175 m	1.162 m	7
FU Campus Long	0.186 m	0.188 m	0.737 m	6
Englerallee	0.140 m	0.139 m	0.507 m	8

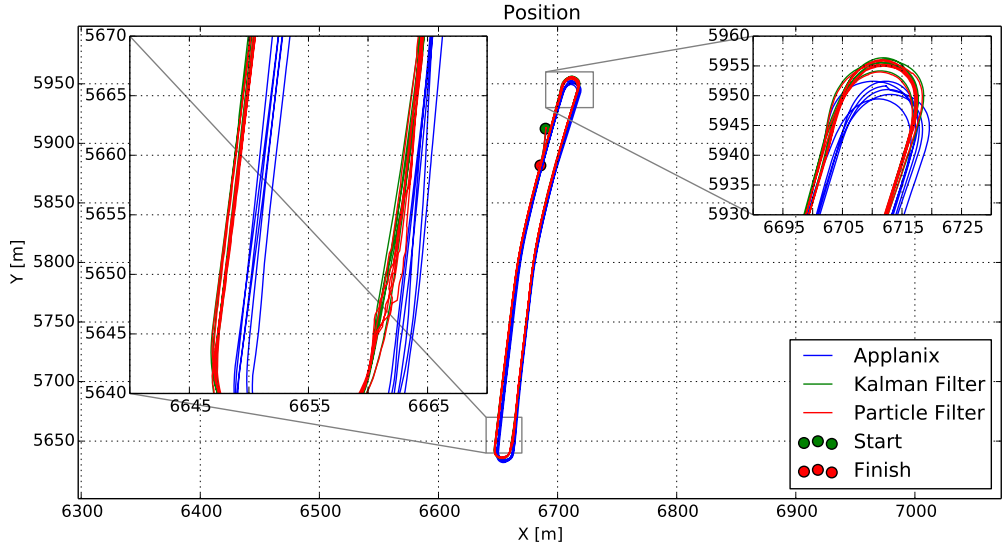
on a course, different localization results can be easily compared without ground truth. Therefore, the position accuracy of the particle filter can be compared to the results of the Output Kalman Filter and the DGPS solution. A drawback of the measure is of course that it only measures consistency, a bias in localization cannot be measured. Despite that, this measure is seen as the most valuable and primary metric to assess the accuracy of the localization solution. The drawbacks of DGPS reference system and the distance to lane markers measure outweigh their advantages.

Table 6.9 shows the repeatability for three tests carried out on the campus and on Englerallee. 50 Monte Carlos runs were averaged to create the results. All three tests result in a repeatability below 0.2 m. The shorter laps in the Englerallee setting might have allowed the driver to drive more accurately than on the campus, resulting in a slightly better result. As the error during acquisition does not necessarily add to or subtract from the localization error, it can be assumed that real repeatability is below 0.3 m. The Kalman filter seems to have only a marginal influence on the repeatability results, which complies with the design goals to smooth the output, filter outliers and increase the output rate without changing the output more than necessary. The results for the Applanix are considerably higher with a range from 0.5 m to 1.2 m. They can vary noticeably even in very similar scenarios, as was the case for the results for Campus Short vs. Campus Long show.

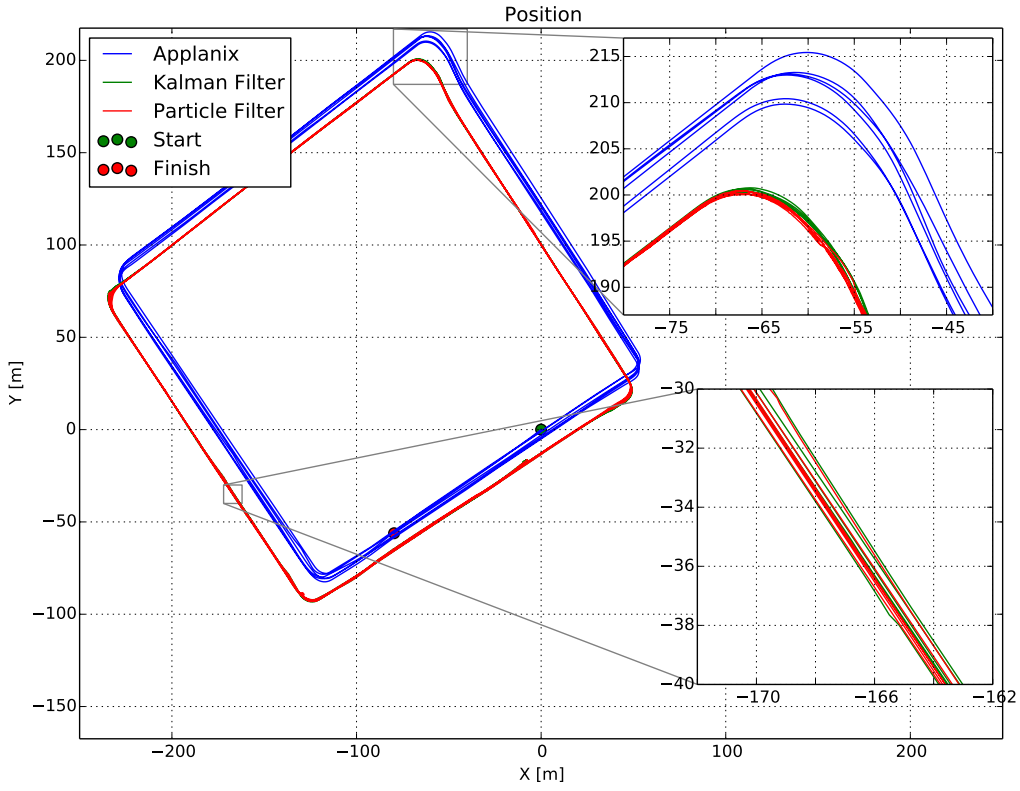
Fig. 6.14 shows plots of the reported locations of the Applanix, the particle filter and the output Kalman filter for two test runs. The particle filter is able to localize the vehicle with a very high repeatability on straight sections and bends, the lateral variation between laps is a lot lower than the Applanix. The Kalman filter smooths the result of the particle filter, as illustrated in Fig. 6.14a on the left, without obvious effects of over-filtering, such as lag or a too high level off stiffness.

Distance to Lane Markers Measure

This accuracy measure relies on the presence of lane markers in the test scenario and an algorithmically independent way of detecting them, e.g. by a lane detection algorithm. The test vehicle has such a system. During mapping, the position of the left and right lane markers of the vehicle ego lane can be mapped as well. In the localization phase, the current distance to the lane markings at the front axle can be subtracted from the distance to the markings in the map. If the markings have not changed between mapping and localization, this is a lateral error, which of course also depends on the



(a) Englerallee: U-turns have to be excluded from the repeatability measure, as the laps were not identical at these points. After the south U-turn the particle filter is obviously somewhat unstable. The Kalman filter stays on a smooth course.



(b) Campus Short: On straight parts and in the corners the localization performance is good. The offset to the Applanix position is due to an offset between mapping time and test time. Applanix drift is approximately 2 m over all laps.

Figure 6.14: Example repeatability for test runs, with close-ups for two routes

accuracy of the lane detection system (Fig. 6.15). Unfortunately, the lane detection in the test carrier was developed with a focus on rural roads and highways and speeds higher than 50 km/h and does not deliver a very high lateral precision in urban settings. Furthermore, all errors made during the mapping step, like the DGPS inaccuracies, have an effect on the lateral errors. Nevertheless, it is interesting to compare the results with the repeatability measure.

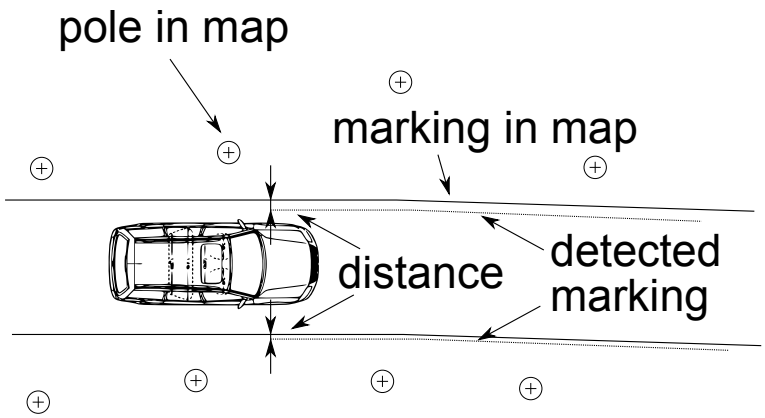


Figure 6.15: Distance to lane markers measure - sketch of the idea

Using the same recording of the Englerallee test route as in Table 6.9, we get a mean lateral offset to the markings of -0.043 m and a standard deviation of 0.146 m. This confirms the repeatability measurements and shows that the performance is comparable to human drivers and sufficient for autonomous driving.

Number of Re-Initializations

During the test runs, the particle filter can get lost without the possibility to recover on its own. How can this be detected without ground truth? A threshold on the position uncertainty of the filter can be used. If the geometric mean of the particle filter position standard deviation is above 15 m, the filter is considered lost and a re-initialization is triggered. The number of those re-initializations can be counted and used to compare parameter sets.

6.4.2 Influence of Frame Rate on Localization

Using a recording of several laps on the Campus Short test route, the influence of the processing frame rate on the localization was tested (Fig. 6.16). The results of 50 Monte Carlo runs of the particle filter were averaged to obtain the mean and standard deviation of the localization repeatability. The map used was created using a recording taken approximately one month earlier. Even with only a frame rate of 11 Hz, the average repeatability does not increase much in comparison to the full frame rate of 22 Hz. However, the probability for localization failures is increased, as during the 50 test

runs with 11 Hz, one complete localization error was encountered that the filter could not recover from. This run was excluded as an outlier from the statistics in the plot.

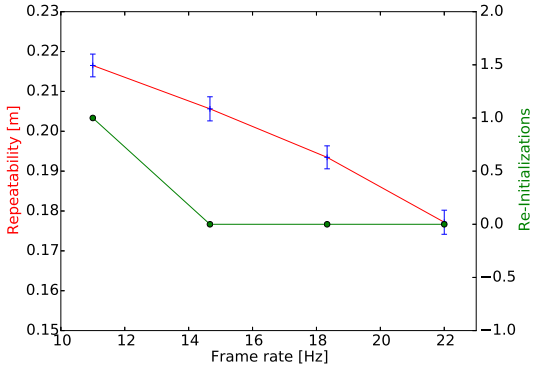


Figure 6.16: Processing frame rate dependency of localization quality quantified by the repeatability measure

6.4.3 Particle Filter Parameter Influence

Number of Particles

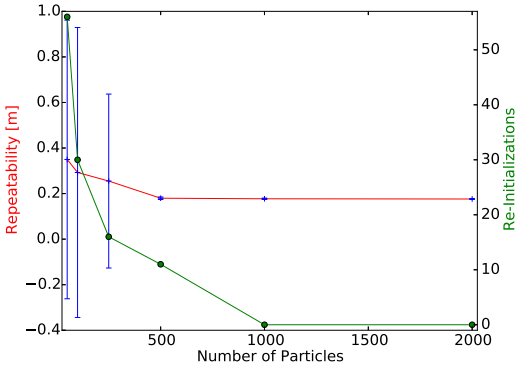
One central question is that regarding the number of particles that is required to enable a reliable function of the localization approach with a particle filter. The number of particles is varied from 50 to 2000. The results show only a minor influence on average repeatability (Fig. 6.17a), but a larger influence on the number of re-initializations. Particle counts below 500 do not seem advisable for the localization task, as the filter gets lost too often. One thousand particles seem to be a good compromise between processing load and robustness.

Likelihood Parameter β_p

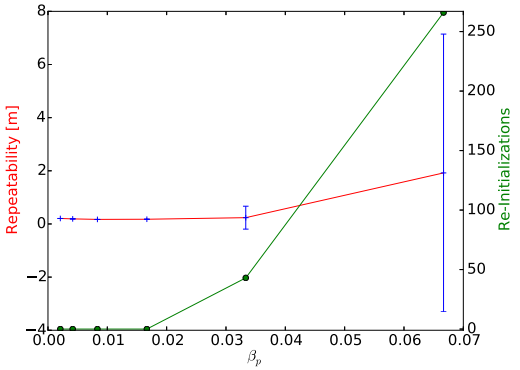
Fig. 6.17b illustrates the effects of varying the parameter β_p of the likelihood, which controls the influence of the distance between matched poles. Track losses rise drastically, if its value is above 0.02. Below, we see no track losses and only a very small effect on repeatability.

Prediction Parameters

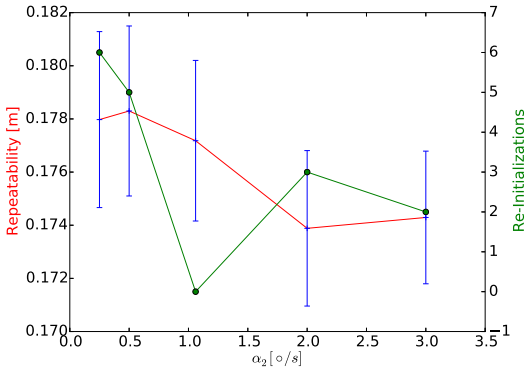
Fig. 6.17c shows that there is no sizable influence from the variance of the yaw rate, controlled by parameter α_2 during prediction. A larger variance leads to a slightly better repeatability, but if the variance is set too high, the number of re-initializations rises. The uncertainty of the filter rises too much and it is not able to cope with short intervals of ambiguous measurements and gets lost. In terms of robustness, a value near $1°/s$ for α_2 seems to be favorable, as the filter did not get lost at this parameter setting.



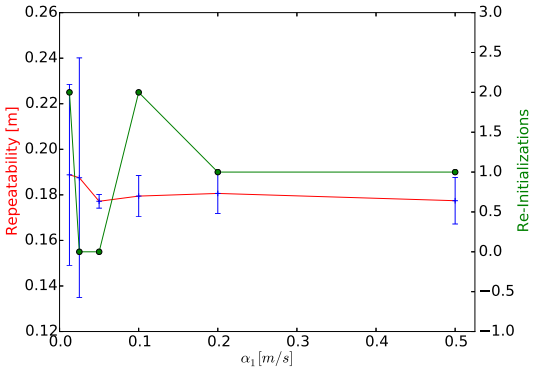
(a) Varying the number of particles



(b) Varying the Mahalanobis distance factor β_p in the likelihood calculation



(c) Varying the yaw rate uncertainty α_2 in the particle filter prediction



(d) Varying the speed uncertainty α_1 in the particle filter prediction

Figure 6.17: Influence of particle filter parameters on repeatability and number of re-initializations

Table 6.10: Kidnapped robot test: On the Englerallee test route statistics were taken from the aggregation of 10 runs. Position changes are tolerated more than orientation changes.

Parameters Test	**Particle Filter**	**#Trials Robots**	**#Lost**	**Mean Return time**
$r_k = 5\,m, \sigma_k = 1°$	$p_r = 0.025, \sigma_r = 2°$	305	2	6.0 s
$r_k = 5\,m, \sigma_k = 5°$	$p_r = 0.025, \sigma_r = 2°$	307	51	8.0 s
$r_k = 5\,m, \sigma_k = 5°$	$p_r = 0.05, \sigma_r = 5°$	303	32	7.1 s

Fig. 6.17d depicts the effect of varying the speed variance controlled by parameter α_1. Too small values lead to a worse repeatability, as the uncertainties of the odometry cannot be compensated any more, a value of more than 0.2 m/s has negative effects as well, as the filter can get lost. This parameter is less critical than α_2, as the number of re-initializations stays low even at extreme parameter settings.

Kidnapped robot test

For the kidnapped robot test, the particle filter is relocated to a position with a change in position determined by a uniform density with radius r_k and an angle standard deviation σ_k. This is a weak form of the test, as relocation is limited to a radius. Kidnapping is modeled as a Poisson process with intensity $\kappa = 0.05\,s$. The robot is considered lost, if its position is at a distance of more than 10 m from the Applanix. In this case, a re-init is triggered. We now measure the number of lost robots per kidnappings and the mean return time. The vehicle is considered to be back in a "returned" position, if the distance to the mean trajectory (without kidnapping) is less than three standard deviations.

Results for Table 6.10 show a very good recovery from kidnappings with position changes. If the angle is changed too much, the return time increases and the loss fractions increase. This can be mitigated by increased exploration with a higher percentage of exploration pixels and a wider angle range.

6.4.4 Execution Times and Latency

During field tests, the system achieved an average processing frame rate of 13.5 Hz, depending on the encountered feature density. Table 6.11 shows that a rather large part of the processing time is needed for the stereo preprocessing, whose runtime remains scene independently constant. This dominates the execution time. Pole detection and tracking and the particle filter vary in their runtimes depending on the number of poles in the scenario and map, but only have a small share of the overall runtime. A three-dimensional view can be rendered with the landmarks, tracked poles, current particles positions and a point cloud to visualize the system state. This visualization needs 5 ms.

The average latency for the pose estimation is 110 ms. This is increased compared to execution times due to additional input/output and the image frame rate of 22 Hz, which causes delays as the processing has to wait till the next frame arrives. The latency is measured for each processed stereo frame pair and used to predict the state estimate accordingly. Odometry is used to continuously predict the output as well.

Table 6.11: Execution times of the complete localization solution on a mobile Intel i7-4960HQ CPU

Stereo Matching	56 ms
Pole Detection & Tracking	8 ms
Particle Filter	5 ms
Output Kalman Filter	< 1 ms
Sum	$\sim$70 ms
Visualization	5 ms

6.4.5 Signal Latency Measurements

In order to react quickly to vehicle maneuvers, the localization output has to follow its dynamics as fast as possible. If the latency in heading or speed changes is too slow, the controller might start to become uncomfortable and maybe oscillate. The time behavior of speed, yaw rate and heading angle can be analyzed using the Applanix as a reference system. Fig. 6.18 shows an example of the Kalman filter output vs. reference. We can cross-correlate the interpolated signals and take the maximum to calculate the latency. Results vary between different recordings, but values are below 1 frame (45 ms) for all three signals. Vehicle speed has the lowest latency with 0-0.4 frames, while the yaw rate has 0.4-0.9 frames. The yaw rate signal could be improved by a model which includes yaw accelerations or the use of the front wheel angle as a predictor for vehicle movement. Yaw angle has a latency of 0-0.8 frames. As the steering and longitudinal control were really smooth during test drives, these values are sufficient for the controller used.

6.4.6 Field Tests

Kleinmachnow First autonomous test drive with pole-based localization, yet without increased update rate of the output Kalman filter.

Campus/Englerallee The campus area was used for first localization and mapping runs. Later, for repeatability tests. The first autonomous test drive with 100 Hz update rate, latency prediction and continuous odometry updates were conducted at Englerallee. We demonstrated successful autonomous driving of multiple laps in a row with challenging U-turns and very good repeatability and smooth lateral and longitudinal control.

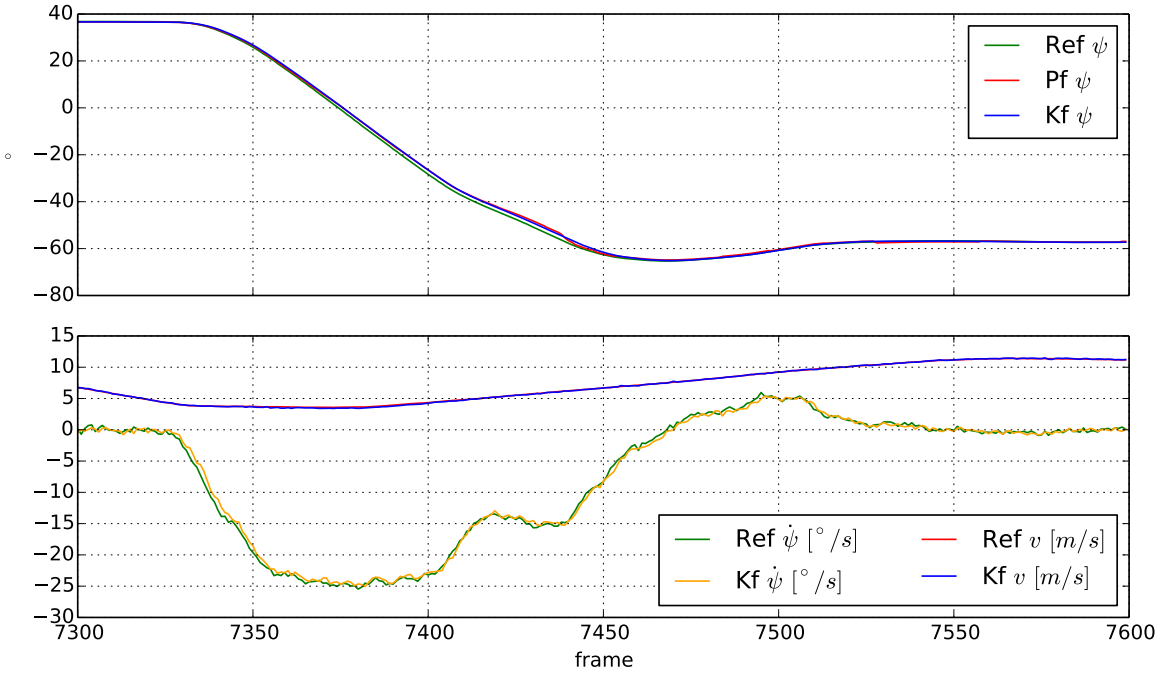


Figure 6.18: Signal Latency - Plot for one turn in the Campus Short test route, Applanix signals are prefixed with Ref, Particle Filter signals with Pf and Kalman Filter signals with Kf. There is no obvious latency in the ψ for the Particle Filter and the Kalman Filter v signal, while the yaw rate $\dot{\psi}$ shows a slight delay. The Kalman filtered yaw lags a little.

Straße des 17. Juni/Brandenburg Gate Autonomous driving was tested on the Straße des 17. Juni, as well, driving on the right and center track of three lane road after one previous mapping run. Driving in the center lane is a very difficult scenario, as the landmarks are very far away and can only be seen for short periods of time. At the test area Reichstag, we drove three laps autonomously in a very difficult scenario, with lots of pedestrians and cyclists. The performance was nearly always good, but some interventions were necessary. One severe jump in localization by the particle filter at a very sparsely populated area of the route lead to the inclusion of the validation gating approach in the Kalman Filter (see 5.3.4). The overall assessment was nevertheless positive, as the system worked even in this very challenging situation.

6.5 Chapter Summary

At first, contributions to automotive stereo matching are evaluated. The proposed stereo online calibration is able to reach sub-pixel accuracy for the relative pitch and roll angle. The new sparse matching costs Center-Symmetric Census Transform shows a comparable accuracy such as that of non-sparse versions of the same size. The weighted variants of the cost even show improvements in matching quality on the KITTI benchmark.

Weighted Semi-Global Matching leads to improvements at certain, single frames with larger, mismatched areas. The probability of making such errors is reduced by using a cost propagation that is driven by the three-dimensional shape around error-prone less textured areas. Rapid Semi-Global Matching is a step forward to parallelize the algorithm with nearly no impact on matching quality, as shown on the KITTI benchmark, achieving the processing speed needed to test driver assistance systems in a prototypical way on CPUs.

In the second part, landmark mapping and localization are focused on. Maps for several test areas are created and presented. Descriptive statistics allow assessing the complexity of the localization task in these areas. It is shown that the Applanix INS/DGPS does not offer the needed accuracy in order to evaluate position information. Therefore, three measures are defined to quantify the localization accuracy: repeatability, distance to lane markers in the map and the number of re-initializations. Repeatability of the localization solution is below 0.2 m on all three test runs in different areas. Hereafter, the influence of several key parameters of the Localization Particle Filter is studied, giving hints to optimal parametrization. A soft version of the kidnapped robot test shows that the localization is robust towards changes in position and small changes in orientation. Latency is below 45 ms for all estimated signals. These are position and orientation information, vehicle speed and yaw rate. In the end, several field tests show that the solution is able to localize an autonomous car reliably with smooth lateral and longitudinal control in different urban scenarios.

7 Conclusion

Accurate and robust localization is one of the key components needed for autonomous driving. One relevant question is how we scan the sensor data for parts that support us in the localization task and which sensor combinations are sufficient. This work has discussed the implications and foundations of using stereo vision for the extraction of landmarks and the localization based on those features in urban scenarios.

Reliable disparity maps calculated from stereo image pairs provide a healthy footing for landmark extraction. As existing stereo matching approaches were either not robust enough or too resource-demanding, chapter 3 dealt with several algorithmic improvements and approaches needed to deliver accurate and robust depth information at high processing speeds. Emphasis was put on sufficient angular resolution of the results to detect even small structures and on a small surface reconstruction error. Matching cost based online calibration is the first key component presented. Then, a new cost-measure is derived by transfer of ideas from LBPs to the Census transform. This cost measurement is more compact and allows the smearing effects caused by larger masks to diminish through the introduction of weights. The robustness of SGM is increased by scene geometry-driven weighting of optimization paths. Finally, we examined changes to SGM that enable real-time performance on CPUs, namely striping and disparity compression.

Chapter 4 is dedicated to the use of pole-like landmarks as a primary feature in urban environments. At first, mainly trees were targeted, but it became obvious that poles from traffic signs, bollards and lamp-posts could also be detected reliably. We developed a method to detect, track and map the full variety of pole-like landmarks. This increased the coverage of this landmark class in urban scenarios considerably. These landmarks were integrated in Chapter 5 in a complete localization solution, using only stereo vision, vehicle odometry and a low-cost GPS. The solution is real-time capable and outputs pose and pose change at 100 Hz. It obeys smoothness requirements and needs only a few landmarks with lightweight attributes to give accurate results.

A thorough evaluation of the presented approaches is given in Chapter 6. The proposed stereo vision algorithms were compared on the KITTI benchmark, resulting in

state-of-the-art results in the class of SGM based algorithms. Weighted SGM in combination with the Weighted Central-Symmetric Census Transform reduces the error above 3 px to 4.97% from 5.76% of the SGM baseline algorithm on the KITTI test set. Rapid SGM is only marginally worse with an error of 5.03% and one of the few higher-ranking algorithms in the benchmark enabling real-time processing. The execution performance was analyzed in detail as well and found to be superior to previous approaches. It can process VGA image pairs with a disparity range of 128 at 16 Hz. Mapping and localization were tested using several test areas covering typical use cases. The maps created have a small footprint, as only a few landmark attributes have to be stored and the needed landmark density for robust localization is also small. This method exhibits an accuracy clearly better than the currently used reference DGPS/INS system and achieves a repeatability below 0.2 m, which is at the same level as a human driver. Successful field tests in a wide range of challenging urban scenarios completed the picture. Even in scenarios with very low landmark densities, the system worked reliably, making clear that pole-like structures are a primary landmark class in urban environments.

We can therefore conclude that it has been shown that the thesis statement is true. The beauty of the approach in comparison to previous localization solutions is the combination of a cheap and versatile sensor with a lightweight landmark, which can be reliably detected and has sufficient coverage in urban scenarios.

7.1 Summary of Contributions

This thesis has made several contributions towards localization for autonomous vehicles in the fields of automotive stereo matching and landmark-based localization. The key contributions of this work are:

- For Stereo Matching, the proposed algorithms do depth reconstruction in an accurate, robust and efficient way, with results applicable to localization purposes:
  - Proposal of matching cost based online calibration. In contrast to classical feature matching approaches, this allowed a high reuse of algorithmic parts from dense stereo matching.
  - A new matching cost was proposed, the Central-Symmetric Census Transform, using ideas from Local Binary Patterns. This cost leads to the same descriptiveness as state-of-the-art matching costs with a compact representation. Introducing weights in the cost even achieves slightly better results.
  - Weighted Semi-Global Matching increased the robustness at textureless areas. A prior on the three-dimensional structure of the modeled scene is created. This prior is used to modify the way paths are integrated in the final aggregated costs by assigning more informative paths higher weights.
  - Rapid Semi-Global Matching improved matching speed by exploiting low- and high-level parallelism on CPUs. The influence of a stripe-based execution is studied and corrected by provisioning overlap areas. Disparity sub-sampling

is included with low overhead. All changes lead only to negligible quality loss. With this approach disparity maps suitable for autonomous driving can be computed on normal CPUs.

- For landmark-based Localization using Stereo Vision we defined landmarks that are frequent in urban scenarios, in the long-term are stable and can be detected reliably by stereo vision and used for localization:
  - Pole-like objects are defined as dependable landmarks in urban scenarios. Based on stereo vision disparity maps, a reliable extraction method for them is presented. These landmarks are included in a lightweight map, used later for localization.
  - A localization method for autonomous driving using particle filters is developed. It consists of the complete processing chain from landmark extraction to position estimation to a smooth latency corrected vehicle pose output. Only stereo vision, vehicle odometry and an off-the-shelf GPS are used as input data. A repeatability below 0.2 m is achieved on several test routes and field tests demonstrate that the system is able to locate autonomous vehicles robustly in urban environments.

7.2 Directions for Future Work

Extensions of the presented approaches to increase the availability and accuracy of the localization are:

Landmark generalization The integration of other landmark classes will be necessary to increase the coverage of the localization to all types of roads. This includes standard features like lane and pavement markings or curbs. A general approach to detect distinct three-dimensional landmarks is beneficial as well, from simple plane-like objects like guard rails and walls to more complex ones like tunnels or arbitrary traffic lights.

Accurate Mapping The localization solution profits from highly accurate mapping as well. A SLAM approach integrating odometry, uncoupled DGPS information and visual odometry for high precision maps seems to be the most promising approach.

Map maintenance Landmarks change over time. Even pole-like landmarks, that are more constant than for example lane markings, will change. Sooner or later landmarks will vanish, have a modified appearance or new ones may appear. Map validation and how online mapping and map updates can be performed for large outdoor environments is still an open question.

Bibliography

[1] M. Agrawal, K. Konolige, and M. Blas. "CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching". English. In: *Computer Vision – ECCV 2008*. Ed. by D. Forsyth, P. Torr, and A. Zisserman. Vol. 5305. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 102–115. DOI: 10.1007/978-3-540-88693-8_8.

[2] E. Alpaydin. *Introduction to Machine Learning*. 2nd. The MIT Press, 2010.

[3] A. Ansar, A. Castano, and L. Matthies. "Enhanced Real-time Stereo Using Bilateral Filtering". In: *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*. Sept. 2004, pp. 455–462. DOI: 10.1109/TDPVT.2004.1335273.

[4] H. Badino, D. Huber, and T. Kanade. "Visual Topometric Localization". In: *Intelligent Vehicles Symposium (IV)*. Baden-Baden, Germany, June 2011.

[5] H. Badino, U. Franke, and D. Pfeiffer. "The Stixel World - A Compact Medium Level Representation of the 3D-World". In: *Pattern Recognition*. Ed. by J. Denzler, G. Notni, and H. Süße. Vol. 5748. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 51–60. DOI: 10.1007/978-3-642-03798-6_6.

[6] C. Banz, P. Pirsch, and H. Blume. "Evaluation of Penalty Functions For Semi-Global Matching Cost Aggregation". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXIX-B3 (2012), pp. 1–6. DOI: 10.5194/isprsarchives-XXXIX-B3-1-2012.

[7] Y. Bar-Shalom and X. R. Li. *Estimation and Tracking: Principles, Techniques and Software*. Artech House, Norwood, MA, 1993.

[8] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li. *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley and Sons, Inc., 2002.

[9] H. Bay et al. “Speeded-Up Robust Features (SURF)”. In: *Comput. Vis. Image Underst.* 110.3 (2008), pp. 346–359. DOI: 10.1016/j.cviu.2007.09.014.

[10] Z. Ben Azouz, C. Shu, and A. Mantel. “Automatic Locating of Anthropometric Landmarks on 3D Human Models”. In: *3DPVT ’06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT’06)*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 750–757. DOI: 10.1109/3DPVT.2006.34.

[11] P. Bender, J. Ziegler, and C. Stiller. “Lanelets: Efficient map representation for autonomous driving”. In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. June 2014, pp. 420–425. DOI: 10.1109/IVS.2014.6856487.

[12] J. Bigün and G. H. Granlund. “Optimal Orientation Detection of Linear Symmetry”. In: *Proceedings of the IEEE First International Conference on Computer Vision*. London, Great Britain, June 1987, pp. 433–438.

[13] S. Birchfield and C. Tomasi. “Depth Discontinuities by Pixel-to-Pixel Stereo”. In: *Int. J. Comput. Vision* 35.3 (Dec. 1999), pp. 269–293. DOI: 10.1023/A:1008160311296.

[14] M. Bleyer and M. Gelautz. “A Layered Stereo Matching Algorithm Using Image Segmentation and Global Visibility Constraints”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 59.3 (2005), pp. 128–150.

[15] M. Bleyer and M. Gelautz. “Simple but Effective Tree Structures for Dynamic Programming-based Stereo Matching”. In: *International Conference on Computer Vision Theory and Applications (VISAPP)*. 2008, pp. 415–422.

[16] A. F. Bobick and S. S. Intille. “Large Occlusion Stereo”. English. In: *International Journal of Computer Vision* 33.3 (1999), pp. 181–200. DOI: 10.1023/A:1008150329890.

[17] O. Bousquet and L. Bottou. “The Tradeoffs of Large Scale Learning”. In: *Advances in Neural Information Processing Systems 20*. Ed. by J. Platt et al. Curran Associates, Inc., 2008, pp. 161–168.

[18] Y. Boykov, O. Veksler, and R. Zabih. “Fast Approximate Energy Minimization via Graph Cuts”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.11 (Nov. 2001), pp. 1222–1239. DOI: 10.1109/34.969114.

[19] L. Breiman. “Random Forests”. In: *Mach. Learn.* 45.1 (Oct. 2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[20] L. Breiman et al. *Classification and Regression Trees*. Statistics/Probability Series. Belmont, California, U.S.A.: Wadsworth Publishing Company, 1984.

[21] C. Brenner. “Global Localization of Vehicles Using Local Pole Patterns”. English. In: *Pattern Recognition*. Ed. by J. Denzler, G. Notni, and H. Süße. Vol. 5748. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 61–70. DOI: 10.1007/978-3-642-03798-6_7.

[22] M. Buder. “Dense real-time stereo matching using memory efficient semi-global-matching variant based on FPGAs”. In: vol. 8437. 2012, DOI: 10.1117/12.921147.

[23] J. Carlson. “Mapping Large, Urban Environments with GPS-Aided SLAM”. PhD thesis. Carnegie Mellon University, 2010.

[24] G. Cerutti et al. “Curvature-Scale-based Contour Understanding for Leaf Margin Shape Recognition and Species Identification”. In: *VISAPP 2013 - Proceedings of the International Conference on Computer Vision Theory and Applications, Volume 1, Barcelona, Spain, 21-24 February, 2013.* Ed. by S. Battiato and J. Braz. SciTePress, 2013, pp. 277–284.

[25] D. Comaniciu and P. Meer. “Mean Shift: A Robust Approach Toward Feature Space Analysis”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.5 (May 2002), pp. 603–619. DOI: 10.1109/34.1000236.

[26] T. Dang, C. Hoffmann, and C. Stiller. “Continuous Stereo Self-Calibration by Camera Parameter Tracking”. In: *Trans. Img. Proc.* 18.7 (July 2009), pp. 1536–1550. DOI: 10.1109/TIP.2009.2017824.

[27] H. Dodel and D. Häupler. *Satellitennavigation.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[28] A. Doucet and A. M. Johansen. “A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later”. In: *The Oxford Handbook of Nonlinear Filtering.* Ed. by D. Crisan and B. Rozovsky. Oxford University Press, 2009.

[29] I. Ernst and H. Hirschmüller. “Mutual Information Based Semi-Global Stereo Matching on the GPU”. In: *Proceedings of the 4th International Symposium on Advances in Visual Computing.* ISVC ’08. Las Vegas, NV: Springer-Verlag, 2008, pp. 228–239. DOI: 10.1007/978-3-540-89639-5_22.

[30] M. Fritz, B. Schiele, and J. H. Piater, eds. *Computer Vision Systems, 7th International Conference on Computer Vision Systems, ICVS 2009, Liège, Belgium, October 13-15, 2009, Proceedings.* Vol. 5815. Lecture Notes in Computer Science. Springer, 2009.

[31] S. Gehrig and U. Franke. “Improving Stereo Sub-Pixel Accuracy for Long Range Stereo”. In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on.* Oct. 2007, pp. 1–7. DOI: 10.1109/ICCV.2007.4409212.

[32] S. K. Gehrig, F. Eberli, and T. Meyer. “A Real-Time Low-Power Stereo Vision Engine Using Semi-Global Matching”. In: *ICVS.* Ed. by M. Fritz, B. Schiele, and J. H. Piater. Vol. 5815. Lecture Notes in Computer Science. Springer, 2009, pp. 134–143.

[33] S. K. Gehrig and C. Rabe. “Real-Time Semi-Global Matching on the CPU”. In: *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshops.* San Francisco, CA, USA, June 2010, pp. 85–92.

[34] A. Geiger, P. Lenz, and R. Urtasun. "Are we ready for autonomous driving? The KITTI vision benchmark suite". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. June 2012, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074.

[35] A. Geiger, M. Roser, and R. Urtasun. "Efficient Large-Scale Stereo Matching". English. In: *Computer Vision – ACCV 2010*. Ed. by R. Kimmel, R. Klette, and A. Sugimoto. Vol. 6492. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 25–38. DOI: 10.1007/978-3-642-19315-6_3.

[36] A. Geiger et al. "Team AnnieWAY's Entry to the 2011 Grand Cooperative Driving Challenge". In: *Intelligent Transportation Systems, IEEE Transactions on* 13.3 (Sept. 2012), pp. 1008–1017. DOI: 10.1109/TITS.2012.2189882.

[37] T. D. Gillespie. *Fundamentals of Vehicle Dynamics*. Society of Automotive Engineers, 1992.

[38] D. Göhring. *Controller Architecture for the Autonomous Cars: MadeInGermany and e-Instein*. Tech. rep. Projekt AutoNOMOS, Freie Universität Berlin, 2012.

[39] D. Gruyer, R. Belaroussi, and M. Revilloud. "Map-aided localization with lateral perception". In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. June 2014, pp. 674–680. DOI: 10.1109/IVS.2014.6856528.

[40] I. Guyon et al. "Gene Selection for Cancer Classification using Support Vector Machines". English. In: *Machine Learning* 46.1-3 (2002), pp. 389–422. DOI: 10.1023/A:1012487302797.

[41] K. He, J. Sun, and X. Tang. "Guided Image Filtering". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 99.PrePrints (2012), p. 1. DOI: 10.1109/TPAMI.2012.213.

[42] M. Heikkilä, M. Pietikäinen, and C. Schmid. "Description of interest regions with local binary patterns". In: *Pattern Recogn.* 42.3 (Mar. 2009), pp. 425–436. DOI: 10.1016/j.patcog.2008.08.014.

[43] J. L. Hennessy and D. A. Patterson. *Computer Architecture, Fifth Edition: A Quantitative Approach*. 5th. Elsevier, 2014.

[44] S. Hermann and R. Klette. "Iterative Semi-global Matching for Robust Driver Assistance Systems". In: *Proceedings of the 11th Asian Conference on Computer Vision - Volume Part III*. ACCV'12. Daejeon, Korea: Springer-Verlag, 2013, pp. 465–478. DOI: 10.1007/978-3-642-37431-9_36.

[45] H. Hirschmüller. "Stereo Processing by Semiglobal Matching and Mutual Information". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 30.2 (2008), pp. 328–341.

[46] H. Hirschmüller. "Stereo Vision in Structured Environments by Consistent Semi-Global Matching". In: *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2386–2393. DOI: 10.1109/CVPR.2006.294.

[47] H. Hirschmüller and S. K. Gehrig. "Stereo matching in the presence of sub-pixel calibration errors". In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. June 2009, pp. 437–444. DOI: 10.1109/CVPR.2009.5206493.

[48] H. Hirschmüller, P. R. Innocent, and J. M. Garibaldi. "Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics". In: *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on*. Vol. 2. Dec. 2002, 1099–1104 vol.2. DOI: 10.1109/ICARCV.2002.1238577.

[49] H. Hirschmüller and D. Scharstein. "Evaluation of Cost Functions for Stereo Matching". In: *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*. June 2007, pp. 1–8. DOI: 10.1109/CVPR.2007.383248.

[50] H. Hirschmüller and D. Scharstein. "Evaluation of Stereo Matching Costs on Images with Radiometric Differences". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 31.9 (2009), pp. 1582–1599.

[51] H. Ho and D. Gibbins. "Curvature-based approach for multi-scale feature extraction from 3D meshes and unstructured point clouds". In: *Computer Vision, IET* 3.4 (Dec. 2009), pp. 201–212. DOI: 10.1049/iet-cvi.2009.0044.

[52] A. Hornung et al. "OctoMap: an efficient probabilistic 3D mapping framework based on octrees". In: *Auton. Robots* 34.3 (2013), pp. 189–206. DOI: 10.1007/s10514-012-9321-0.

[53] A. Howard. "Real-time stereo visual odometry for autonomous ground vehicles". In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. Sept. 2008, pp. 3946–3952. DOI: 10.1109/IROS.2008.4651147.

[54] X. Hu and P. Mordohai. "A Quantitative Evaluation of Confidence Measures for Stereo Vision". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (2012), pp. 2121–2133. DOI: 10.1109/TPAMI.2012.46.

[55] M. Humenberger, T. Engelke, and W. Kubinger. "A census-based stereo vision algorithm using modified Semi-Global Matching and plane fitting to improve matching quality". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. June 2010, pp. 77–84. DOI: 10.1109/CVPRW.2010.5543769.

[56] C. Im, H. Nishida, and T. L. Kunii. "Recognizing Plant Species by Normalized Leaf Shapes". In: *Vision Interface*. 1999.

[57] O. Irsoy, O. T. Yildiz, and E. Alpaydin. "Design and Analysis of Classifier Learning Experiments in Bioinformatics: Survey and Case Studies". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 9.6 (2012), pp. 1663–1675. DOI: 10.1109/TCBB.2012.117.

[58] A. E. Johnson and M. Hebert. "Using spin images for efficient object recognition in cluttered 3D scenes". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 21.5 (May 1999), pp. 433–449. DOI: 10.1109/34.765655.

[59] R. Jonker and A. Volgenant. "A shortest augmenting path algorithm for dense and sparse linear assignment problems". English. In: *Computing* 38.4 (1987), pp. 325–340. DOI: 10.1007/BF02278710.

[60] S. Julier, J. Uhlmann, and H. Durrant-Whyte. "A new method for the nonlinear transformation of means and covariances in filters and estimators". In: *Automatic Control, IEEE Transactions on* 45.3 (Mar. 2000), pp. 477–482. DOI: 10.1109/9.847726.

[61] A. Kelly. *A 3D Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles*. Tech. rep. CMU-RI-TR-94-19. Pittsburgh, PA: Robotics Institute, May 1994.

[62] W. Khan et al. "Belief Propagation stereo matching compared to iSGM on binocular or trinocular video data". In: *Intelligent Vehicles Symposium (IV), 2013 IEEE*. 2013, pp. 791–796. DOI: 10.1109/IVS.2013.6629563.

[63] B. Kitt, A. Geiger, and H. Lategahn. "Visual Odometry based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme". In: *IEEE Intelligent Vehicles Symposium*. San Diego, USA, June 2010.

[64] K. Konolige, M. Agrawal, and J. Solà. "Large scale visual odometry for rough terrain". In: *In Proc. International Symposium on Robotics Research*. 2007.

[65] S. Kramm, P. Miche, and A. Bensrhair. "Self calibration of a road stereo vision system through correlation criterions". In: *IEEE Intelligent Vehicles Symposium*. 2006, pp. 36–41.

[66] J. Levinson and S. Thrun. "Robust Vehicle Localization in Urban Environments Using Probabilistic Maps". In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. May 2010, pp. 4372–4378. DOI: 10.1109/ROBOT.2010.5509700.

[67] D. G. Lowe. "Distinctive image features from scale-invariant keypoints". In: *International Journal of Computer Vision* 60 (2004), pp. 91–110.

[68] M. Lundgren et al. "Vehicle self-localization using off-the-shelf sensors and a detailed map". In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. June 2014, pp. 522–528. DOI: 10.1109/IVS.2014.6856524.

[69] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[70] J. Matas et al. "Robust wide-baseline stereo from maximally stable extremal regions". In: *Image Vision Comput.* 22.10 (2004), pp. 761–767.

[71] L. Matthies and S. Shafer. "Error modeling in stereo navigation". In: *Robotics and Automation, IEEE Journal of* 3.3 (June 1987), pp. 239–248. DOI: 10.1109/JRA.1987.1087097.

[72] S. Mattoccia, F. Tombari, and L. Di Stefano. “Stereo Vision Enabling Precise Border Localization Within a Scanline Optimization Framework”. English. In: *Computer Vision ,Äì ACCV 2007*. Ed. by Y. Yagi et al. Vol. 4844. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, pp. 517–527. DOI: 10.1007/978-3-540-76390-1_51.

[73] G. V. Meerbergen et al. “A Hierarchical Symmetric Stereo Algorithm Using Dynamic Programming.” In: *International Journal of Computer Vision* 47.1-3 (2002), pp. 275–285.

[74] C. Mei et al. “A Constant-Time Efficient Stereo SLAM System”. In: *Proceedings of the British Machine Vision Conference*. BMVA Press, 2009, pp. 54.1–54.11. DOI: 10.5244/C.23.54.

[75] M. Michael et al. “Real-time stereo vision: Optimizing Semi-Global Matching”. In: *Intelligent Vehicles Symposium (IV), 2013 IEEE*. 2013, pp. 1197–1202. DOI: 10.1109/IVS.2013.6629629.

[76] K. Mikolajczyk and C. Schmid. “A Performance Evaluation of Local Descriptors”. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 27.10 (2005), pp. 1615–1630.

[77] M. Montemerlo et al. “FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping That Provably Converges”. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence*. IJCAI’03. Acapulco, Mexico: Morgan Kaufmann Publishers Inc., 2003, pp. 1151–1156.

[78] M. Montemerlo et al. “FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem”. In: *In Proceedings of the AAAI National Conference on Artificial Intelligence*. AAAI, 2002, pp. 593–598.

[79] W. Muła. *SSSE3: Fast popcount (accessed 4.1.2014)*. Apr. 2010. URL: http://wm.ite.pl/articles/sse-popcount.html.

[80] J. Munkres. “Algorithms for the Assignment and Transportation Problems”. In: *Journal of the Society of Industrial and Applied Mathematics* 5.1 (Mar. 1957), pp. 32–38.

[81] D. Nister, O. Naroditsky, and J. Bergen. “Visual Odometry”. In: *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* 1 (2004), pp. 652–659. DOI: 10.1109/CVPR.2004.265.

[82] T. Ojala, M. Pietikainen, and T. Maenpaa. “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.7 (Jul), pp. 971–987. DOI: 10.1109/TPAMI.2002.1017623.

[83] E. Olson. “Real-time correlative scan matching”. In: *Robotics and Automation, 2009. ICRA ’09. IEEE International Conference on*. 2009, pp. 4387–4393. DOI: 10.1109/ROBOT.2009.5152375.

[84] *OpenMP C and C++ Application Program Interface Version 2.0*. Tech. rep. OpenMP Architecture Review Board, Mar. 2002.

[85] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[86] D. Pfeiffer and U. Franke. "Efficient representation of traffic scenes by means of dynamic stixels". In: *Intelligent Vehicles Symposium (IV), 2010 IEEE*. 2010, pp. 217–224. DOI: 10.1109/IVS.2010.5548114.

[87] D. Pfeiffer and U. Franke. "Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data". In: *British Machine Vision Conference (BMVC)*. Dundee, Scotland, Aug. 2011.

[88] O. Pink, F. Moosmann, and A. Bachmann. "Visual Features for Vehicle Localization and Ego-Motion Estimation". In: *Intelligent Vehicles Symposium, 2009 IEEE*. June 2009, pp. 254–260. DOI: 10.1109/IVS.2009.5164287.

[89] O. Pink. "Bildbasierte Selbstlokalisierung von Straßenfahrzeugen". PhD thesis. Institut für Mess- und Regelungstechnik mit Maschinenlaboratorium (MRT) , KIT, 2011.

[90] *POS LV V4 Installation and Operation Guide*. PUBS-MAN-000048, Revision 5. Applanix. June 2008.

[91] *POS LV V4 User Control, Display, Data and Logging Port Interface Control Document*. PUBS-ICD-000036, Revision 5. Applanix. June 2009.

[92] J. R. Quinlan. "Induction of Decision Trees". In: *Mach. Learn.* 1.1 (Mar. 1986), pp. 81–106. DOI: 10.1023/A:1022643204877.

[93] K. Ramm. "Evaluation von Filter-Ansätzen für die Positionsschätzung von Fahrzeugen mit den Werkzeugen der Sensitivitätsanalyse". ger. PhD thesis. Universität Stuttgart, 2008.

[94] E. Rosten and T. Drummond. "Fusing points and lines for high performance tracking." In: *IEEE International Conference on Computer Vision*. Vol. 2. Oct. 2005, pp. 1508–1511. DOI: 10.1109/ICCV.2005.104.

[95] E. Rosten, R. Porter, and T. Drummond. "Faster and Better: A Machine Learning Approach to Corner Detection". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.1 (Jan. 2010), pp. 105–119. DOI: 10.1109/TPAMI.2008.275.

[96] R. B. Rusu, N. Blodow, and M. Beetz. "Fast Point Feature Histograms (FPFH) for 3D Registration". In: *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*. ICRA'09. Kobe, Japan: IEEE Press, 2009, pp. 1848–1853.

[97] D. Scharstein and R. Szeliski. "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms". English. In: *International Journal of Computer Vision* 47.1-3 (2002), pp. 7–42. DOI: 10.1023/A:1014573219977.

[98] A. Schindler. "Vehicle Self-Localization Using High-Precision Digital Maps". In: *Intelligent Vehicles Symposium (IV), 2013 IEEE*. June 2013, pp. 141–146. DOI: 10.1109/IVS.2013.6629461.

[99] A. Schlichting and C. Brenner. "Localization using automotive laser scanners and local pattern matching". In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. June 2014, pp. 414–419. DOI: 10.1109/IVS.2014.6856460.

[100] R. Schubert, N. Mattern, and G. Wanielik. "An evaluation of nonlinear filtering algorithms for integrating GNSS and inertial measurements". In: *Position, Location and Navigation Symposium, 2008 IEEE/ION*. May 2008, pp. 25–29. DOI: 10.1109/PLANS.2008.4569966.

[101] R. Schubert, E. Richter, and G. Wanielik. "Comparison and evaluation of advanced motion models for vehicle tracking". In: *Information Fusion, 2008 11th International Conference on*. June 2008, pp. 1–6.

[102] M. Shimizu and M. Okutomi. "Sub-Pixel Estimation Error Cancellation on Area-Based Matching". In: *Int. J. Comput. Vision* 63.3 (July 2005), pp. 207–224. DOI: 10.1007/s11263-005-6878-5.

[103] G. Sibley et al. "Planes, trains and automobiles - autonomy for the modern robot". In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. May 2010, pp. 285–292. DOI: 10.1109/ROBOT.2010.5509527.

[104] L. J. Skelly and S. Sclaroff. "Improved feature descriptors for 3D surface matching". In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. Vol. 6762. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Sept. 2007. DOI: 10.1117/12.753263.

[105] R. C. Smith and P. Cheeseman. "On the Representation and Estimation of Spatial Uncertainly". In: *Int. J. Rob. Res.* 5.4 (Dec. 1986), pp. 56–68. DOI: 10.1177/027836498600500404.

[106] R. Spangenberg, T. Langner, and R. Rojas. "On-line Stereo Self-calibration through Minimization of Matching Costs". In: *Image Analysis*. Ed. by J.-K. Kämäräinen and M. Koskela. Vol. 7944. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 545–554. DOI: 10.1007/978-3-642-38886-6_51.

[107] R. Spangenberg, T. Langner, and R. Rojas. "Weighted Semi-Global Matching and Center-Symmetric Census Transform for Robust Driver Assistance". In: *Computer Analysis of Images and Patterns*. Ed. by R. Wilson et al. Vol. 8048. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 34–41. DOI: 10.1007/978-3-642-40246-3_5.

[108] R. Spangenberg et al. "Large scale Semi-Global Matching on the CPU". In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. June 2014, pp. 195–201. DOI: 10.1109/IVS.2014.6856419.

[109] B. Steder et al. “NARF: 3D Range Image Features for Object Recognition”. In: *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. Taipei, Taiwan, Oct. 2010.

[110] P. Steingrube, S. K. Gehrig, and U. Franke. “Performance Evaluation of Stereo Algorithms for Automotive Applications.” In: *ICVS*. Ed. by M. Fritz, B. Schiele, and J. H. Piater. Vol. 5815. Lecture Notes in Computer Science. Springer, Oct. 15, 2009, pp. 285–294.

[111] J. Sun, N.-N. Zheng, and H.-Y. Shum. “Stereo Matching Using Belief Propagation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 25.7 (July 2003), pp. 787–800. DOI: 10.1109/TPAMI.2003.1206509.

[112] The Orocos Project. *Orocos Project Web Site*. Mar. 2015. URL: http://www.orocos.com.

[113] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[114] C. Tomasi and R. Manduchi. “Bilateral Filtering for Gray and Color Images”. In: *Proceedings of the Sixth International Conference on Computer Vision*. ICCV ’98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839+.

[115] C. Valgren and A. J. Lilienthal. “SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments”. In: *Robotics and Autonomous Systems* (Sept. 2009). DOI: 10.1016/j.robot.2009.09.010.

[116] B.-T. Vo and B.-N. Vo. “Labeled Random Finite Sets and Multi-Object Conjugate Priors”. In: *Signal Processing, IEEE Transactions on* 61.13 (July 2013), pp. 3460–3475. DOI: 10.1109/TSP.2013.2259822.

[117] P. Waldmann. “Entwicklung eines Fahrzeugführungssystems zum Erlernen der Ideallinie auf Rennstrecken”. PhD thesis. Brandenburgische Technische Universität Cottbus, 2011.

[118] M. Wang. “A Cognitive Navigation Approach for Autonomous Vehicles”. PhD thesis. Freie Universität Berlin, 2012.

[119] J. Wiest et al. “Multi-sensor self-localization based on Maximally Stable Extremal Regions”. In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. June 2014, pp. 555–560. DOI: 10.1109/IVS.2014.6856413.

[120] R. Zabih and J. Woodfill. “Non-parametric Local Transforms for Computing Visual Correspondence”. In: *Proceedings of the Third European Conference-Volume II on Computer Vision - Volume II*. ECCV ’94. Stockholm, Sweden: Springer-Verlag New York, Inc., 1994, pp. 151–158.

[121] Z. Zhang. “A Flexible New Technique for Camera Calibration”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22.11 (2000), pp. 1330–1334.

[122] J. Zhou, H. Peng, and T. Gordon. "Characterization of the Lateral Control Performance by Human Drivers on Highways." In: *SAE Int. J. Passeng. Cars - Mech. Syst.* 1.1 (Apr. 2009), pp. 450–458. DOI: 10.4271/2008-01-0561.

[123] J. Ziegler et al. "Video based localization for Bertha". In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. June 2014, pp. 1231–1238. DOI: 10.1109/IVS.2014.6856560.

[124] C. Zinner et al. "An Optimized Software-Based Implementation of a Census-Based Stereo Matching Algorithm". In: *Proceedings of the 4th International Symposium on Advances in Visual Computing*. ISVC '08. Las Vegas, NV: Springer-Verlag, 2008, pp. 216–227. DOI: 10.1007/978-3-540-89639-5_21.

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich alle Hilfsmittel und Hilfe angegeben habe und auf dieser Grundlage die Arbeit selbstständig verfasst habe.

Ich erkläre weiterhin, dass die Arbeit nicht schon einmal in einem früheren Promotionsverfahren eingereicht wurde.

Berlin, den 28. Oktober 2015

Robert Spangenberg