STEREOSCOPIC METHODS FOR HIGH-PERFORMANCE OBJECT DETECTION AND DISTANCE ESTIMATION

EXTENDING VISUAL ENVIRONMENT PERCEPTION FOR INTELLIGENT VEHICLES

Dissertation

zur Erlangung des Doktorgrades

der Naturwissenschaften

Vorgelegt beim Fachbereich Informatik und Mathematik

der Johann Wolfgang Goethe-Universität

in Frankfurt am Main

von

PETER PINGGERA

aus Hallein

Frankfurt am Main 2018

D 30

STEREOSCOPIC METHODS FOR HIGH-PERFORMANCE OBJECT DETECTION AND DISTANCE ESTIMATION

EXTENDING VISUAL ENVIRONMENT PERCEPTION FOR INTELLIGENT VEHICLES

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

Vorgelegt beim Fachbereich Informatik und Mathematik
der Johann Wolfgang Goethe-Universität
in Frankfurt am Main

von

PETER PINGGERA
aus Hallein

Frankfurt am Main 2018

D 30

Vom Fachbereich Informatik und Mathematik der
Johann Wolfgang Goethe-Universität als Dissertation angenommen.

DEKAN:	Prof. Dr. Andreas Bernig
1. GUTACHTER:	Prof. Dr. Rudolf Mester
2. GUTACHTER:	Prof. Dr. Bastian Leibe
DATUM DER DISPUTATION:	09.05.2018

ABSTRACT

Powerful environment perception systems are a fundamental prerequisite for the successful deployment of intelligent vehicles, from advanced driver assistance systems to self-driving cars. Arguably the most essential task of such systems is the reliable detection and localization of obstacles in order to avoid collisions. Two particularly challenging scenarios in this context are represented by small, unexpected obstacles on the road ahead, and by potentially dynamic objects observed from a large distance. Both scenarios become exceedingly critical when the ego-vehicle is traveling at high speed. As a consequence, two major requirements placed on environment perception systems are the capability of (a) high-sensitivity generic object detection and (b) high-accuracy obstacle distance estimation. The present thesis addresses both requirements by proposing novel approaches based on stereo vision for spatial perception.

First, this work presents a novel method for the detection of small, generic obstacles and objects at long range directly from stereo imagery. The detection is based on sound statistical tests using local geometric criteria which are applicable to both static and moving objects. The approach is not limited to predefined sets of semantic object classes and does not rely on restrictive assumptions on the environment, such as oversimplified global ground surface models. Free-space and obstacle hypotheses are evaluated based on a statistical model of the input image data in order to avoid a loss of sensitivity through intermediate processing steps. In addition to the detection result, the algorithm simultaneously yields refined estimates of object distances, originating from an implicit optimization of the geometric obstacle hypothesis models.

The proposed detection system provides multiple flexible output representations, ranging from 3D obstacle point clouds to compact mid-level obstacle segments to bounding box representations of object instances suitable for model-based tracking. The core algorithm concept lends itself to massive parallelization and can be implemented efficiently on dedicated hardware. Real-time execution is demonstrated on a test vehicle in real-world traffic.

For a thorough quantitative evaluation of the detection performance, two dedicated datasets are employed, covering small and hard-to-detect obstacles in urban environments as well as distant dynamic objects in highway driving scenarios. The proposed system is shown to significantly outperform current general purpose obstacle detection approaches in both setups, providing a considerable increase in detection range while reducing the false positive rate at the same time.

Second, this work considers the high-accuracy estimation of object distances from stereo vision, particularly at long range. Several new methods for optimizing the stereo-based distance estimates of detected objects are proposed and compared to state-of-the-art concepts. A comprehensive statistical evaluation is performed on an extensive dedicated dataset, establishing reference values for the accuracy limits actually achievable in practice. Notably, the refined distance estimates implicitly provided by the proposed obstacle detection system are shown to yield highly accurate results, on par with the top-performing dedicated stereo matching algorithms considered in the analysis.

ZUSAMMENFASSUNG

EINLEITUNG

Leistungsfähige Umgebungserfassungssysteme sind eine wesentliche Voraussetzung für den sicheren und störungsfreien Einsatz von intelligenten Fahrzeugen, von modernen Fahrerassistenzsystemen bis hin zu selbstfahrenden Autos. Die wohl wichtigste Aufgabe eines solchen Systems ist die zuverlässige Erkennung und Lokalisierung von Hindernissen, um rechtzeitig zu reagieren und somit Kollisionen zu vermeiden. Zwei besonders anspruchsvolle Szenarien in diesem Zusammenhang stellen kleine, unerwartete Hindernisse auf der Fahrbahn sowie potenziell bewegte Objekte in großen Entfernungen dar. Beide Szenarien sind umso kritischer, je schneller sich das Eigenfahrzeug bewegt. Daraus ergeben sich zwei wesentliche Anforderungen an Umgebungserfassungssysteme für intelligente Fahrzeuge: (a) die hochsensitive Detektion von generische Objekten bzw. Hindernissen und (b) die hochgenaue Schätzung von Hindernisdistanzen. Die vorliegende Arbeit adressiert beide Anforderungen und präsentiert dazu neue Ansätze basierend auf Stereo-Bildverarbeitung zur räumlichen Wahrnehmung.

OBJEKTDETEKTION

Im ersten Teil der Arbeit wird eine neuartige Methode namens Direct Planar Hypothesis Testing (PHT) zur Detektion von kleinen, generischen Hindernissen und Objekten in großer Distanz durch die direkte Verarbeitung von Stereo-Bilddaten vorgestellt. Der Detektionsansatz beruht auf einer sorgfältigen Analyse der Anforderungen an intelligente Fahrzeuge, welche auf Hindernisse beliebigen Typs auf einer Vielzahl möglicher Straßengeometrien adäquat reagieren müssen. Insbesondere beinhaltet dies die Detektion von sowohl kleinen als auch weit entfernten Objekten. Demnach lassen sich Sensitivität, Flexibilität und Effizienz als wesentliche Kriterien für den Entwurf des Detektionssystems ableiten.

Der vorgestellte Detektionsalgorithmus führt pixelweise binäre Hypothesentests auf kleinen, voneinander unabhängig betrachteten Bildfenstern durch. Freiraum- und Hindernishypothesen werden dabei direkt auf Stereo-Bilddaten bewertet, um einen Sensitivitätsverlust durch zusätzliche Verarbeitungsschritte und Zwischenrepräsentationen zu vermeiden. Das Testergebnis wird jeweils dem zentralen Pixel des untersuchten Bildfensters zugewiesen. Aufgrund der lokalen Betrachtung kann eine zuverlässige Aussage allerdings nur dann getroffen werden, wenn die zugrundeliegenden Bilddaten ein bestimmtes Mindestmaß an Evidenz

ermöglichen. Dies bedingt das Vorhandensein von entsprechender Bildtextur.

Die statistischen Tests beruhen auf geometrischen, lokal planaren Hypothesenmodellen, welche in ihrer Orientierung jeweils um einen vorgegebenen Referenzwert variieren können. Diese Formulierung liefert die notwendige Flexibilität, um auch global schwer modellierbare Bodenoberflächen sowie unterschiedlich geformte Hindernisse abzubilden. Da der Detektionsalgorithmus im Zuge des Hypothesentests implizit eine Optimierung aller geometrischer Hypothesenmodelle direkt auf den Bilddaten durchführt, werden gleichzeitig auch deutlich verbesserte Distanzschätzungen für alle detektierten Objekte erlangt.

Die Formulierung der in PHT verwendeten geometrischen Hypothesenmodelle erfolgt über Ebenen im 3D Raum, was eine sehr flexible Konfiguration erlaubt. Zudem ist die Erweiterung für den Einsatz in kalibrierten Multi-Kamerasystemen problemlos möglich und bietet Potential für eine weitere Erhöhung der Detektionsleistung. Mit Blick auf maximale Effizienz wird jedoch für die vorliegende Anwendung und die eingesetzten Kamerakonfigurationen gezeigt, dass die Anzahl der freien Parameter in der Formulierung des Detektionsproblems ohne Leistungseinbußen weiter reduziert werden kann. So wird eine minimale Parametrisierung des impliziten nichtlinearen Optimierungsproblems im Disparitätsraum erreicht. Diese deutlich effizientere Variante des Detektionssystems wird als Fast Direct Planar Hypothesis Testing (FPHT) bezeichnet.

Durch die unabhängige Analyse lokaler Bildfenster eignen sich sowohl PHT als auch FPHT hervorragend für eine hochgradig parallele Ausführung, was anhand einer entsprechenden Implementierung auf einer GPU demonstriert wird. Darüber hinaus ist der Detektionsalgorithmus sehr gut für eine Portierung auf energieeffiziente dedizierte Hardware wie FPGAs geeignet. Der Echtzeitbetrieb wird in einem Testfahrzeug im realen Straßenverkehr gezeigt und getestet.

Das pixelweise Detektionsergebnis der präsentierten Methoden und die anhand der Dispariätsinformation entstehenden 3D Objektpunktwolken stellen in einem mehrstufigen Umgebungserfassungssystem oftmals keine optimale Eingangsrepräsentation für nachfolgende Verarbeitungsschritte dar. Aus diesem Grund wird in dieser Arbeit eine kompakte und gleichzeitig flexible Segmentrepräsentation namens Cluster-Stixels (CStix) vorgestellt, die von der etablierten *Stixel-Welt* [Badino et al., 2009, Pfeiffer and Franke, 2011, Schneider et al., 2016] inspiriert ist. In der experimentellen Auswertung zeigen sich die Cluster-Stixels als äußerst geeignete Beschreibung für komplexe städtische Verkehrsszenen. Durch die flexible Darstellung können Objekte beliebiger Form ausreichend genau und doch kompakt beschrieben werden. Zudem werden die Detektionsergebnisse oft sogar durch die Interpolation korrekter Einzeldetektionen verbessert.

Als Ergänzung zu Objektpunktwolken und Cluster-Stixels wird außerdem eine klassische Bounding Box (BB) Repräsentation des Detektionsergebnisses präsentiert, die eine passende Eingangsdarstellung für modellbasierte Trackingverfahren liefert. Durch die zeitliche Filterung kann die Detektionskonfidenz erhöht und vereinzelt auftretende Fehldetektionen effektiv unterdrückt werden. Der wirkungsvolle Einsatz in einem Objekttrackingsystem wird anhand von exemplarischen Autobahnszenarien demonstriert.

In einer umfangreichen Analyse wird das vorgestellte Detektionssystem ausführlich getestet und mit aktuellen und in der Praxis etablierten Referenzverfahren verglichen. Der Fokus der Auswertung wird dabei auf zwei kritische Szenarien gelegt: Die Detektion von kleinen, generischen Hindernissen im komplexen urbanen Umfeld, sowie die Detektion von generischen, weit entfernten Objekten in Autobahnszenarien. Um alle untersuchten Ansätze und die entsprechenden Ausgangsrepräsentationen quantitativ zu bewerten, werden dedizierte, manuell annotierte Datensätze erstellt. Dies ist vor allem für die Betrachtung von kleinen, unerwarteten Hindernissen essenziell, da diese in der Praxis vergleichsweise selten zu beobachten und damit nur mit erheblichem Aufwand in signifikanten Mengen zu erfassen sind. Die Detektionsergebnisse werden auf Pixelebene, Instanzebene und auf Objektebene untersucht, wobei eine allgemeine Auswertung sowie eine Analyse in Abhängigkeit der Objektdistanz erfolgt. In sämtlichen Testkategorien und über alle Ausgangsrepräsentationen hinweg übertreffen dabei sowohl PHT als auch FPHT die Referenzverfahren deutlich. Insbesondere wird durch die vorgestellten Methoden eine wesentlich höhere Detektionsreichweite bei zugleich deutlich niedrigerer Falschpositivrate erzielt.

In der Praxis sollten visuelle Umgebungserfassungssysteme von intelligenten Fahrzeugen die vorgestellten Objektdetektionsmethoden nutzen, um damit existierende, allgemeine 3D Szenenmodelle wie beispielsweise die semantische *Stixel Welt* von Schneider et al. [2016] zu erweitern. Auf diese Art und Weise wird eine ganzheitliche räumliche und semantische Szenenbeschreibung erlangt, welche dem Fahrzeug ein umfassendes Verständnis seiner Umgebung ermöglicht und gleichzeitig die in dieser Arbeit behandelten anspruchsvollen Szenarien berücksichtigt. Großes Potential für zukünftige Leistungsverbesserung birgt dabei die direkte Kombination der hier präsentierten geometrischen Detektionsverfahren mit semantischen Ansätzen, basierend auf modernen Methoden des maschinellen Lernens.

Im zweiten Teil der Arbeit wird die hochgenaue Distanzschätzung von detektierten Objekten auf Basis von Stereo-Bilddaten im Detail behandelt. Da Hindernisse im Fahrkorridor des Eigenfahrzeugs besonders relevant für die Kollisionsvermeidung sind, muss ihre Position und Geschwindigkeit mit größtmöglicher Genauigkeit bestimmt werden. Gleichzeitig wirken sich jedoch Fehler im Disparitätsraum der Stereokamera besonders stark auf die Genauigkeit der resultierenden Distanzschätzungen aus. Aus diesem Grund ist eine optimale sub-pixel genaue Disparitätschätzung in ausgewählten Bildbereichen essenziell. Diese konkrete Anforderung steht im Kontrast zu gängigen Stereo-Matching Benchmarks, in welchen in der Regel die durchschnittliche Disparitätsgenauigkeit über ganze Bilder evaluiert wird.

In der vorliegenden Arbeit werden mehrere neuartige Ansätze zur Optimierung der Disparitätsgenauigkeit für Objektinstanzen vorgestellt und mit aktuellen Verfahren verglichen. Die Menge der untersuchten Algorithmen beinhaltet dabei rein lokale Ansätze wie Local Differential Matching (LDM) und Joint Matching and Segmentation (MSEG), jedoch auch Methoden, die eine globale Optimierung von pixelweisen Kosten im diskreten und kontinuierlichen Raum durchführen. Darüber hinaus wird die robuste Kombination von mehreren unabhängigen, lokalen Beobachtungen in eine einzige Objektdistanzschätzung analysiert. Neben den verschiedenen grundlegenden Algorithmenkonzepten werden auch jeweils wesentliche Basiskomponenten untersucht. Dies beinhaltet unter anderem die Wahl von optimalen Filteroperatoren zur Bestimmung von Bildgradienten sowie unterschiedliche Methoden zur Intensitätsinterpolation.

Um eine aussagekräftige statistische Auswertung und eine systematische Bewertung der Disparitätsgenauigkeit zu ermöglichen, wird der Einsatz von robusten, lageparameterfreien Streuungsmaßen zusätzlich zum klassischen mittleren Disparitätsfehler vorgeschlagen. Dies ermöglicht eine faire Bewertung der statistischen Variabilität der Disparitätsschätzungen, unabhängig von einzelnen Ausreißern und etwaigen systematischen Fehlern. Des Weiteren wird ein neuartiges, objektbasiertes Maß der zeitlichen Variation des Disparitätsfehlers eingeführt. Eine geringe zeitliche Fehlervariation ist unter anderem wesentlich für die zuverlässige Bestimmung von Objektgeschwindigkeiten.

Die Studie wird auf einem umfangreichen, dedizierten Datensatz durchgeführt, wobei Objekte in einem Entfernungsbereich von 50 m bis 160 m berücksichtigt werden. Ein Langstrecken-RADAR dient hierbei als Referenzsensor. Sämtliche Leistungsmaße werden sowohl über den gesamten Datensatz als auch in Abhängigkeit der Objektentfernung ausgewertet.

Es zeigt sich, dass die insgesamt höchste Disparitätsgenauigkeit durch die robuste Kombination von mehreren unabhängigen Beobachtungen in eine einzelne Objektdistanzschätzung erzielt wird. Insbesondere die Kombination von lokalen differenziellen Disparitätsschätzern liefert auf diese Art und Weise die besten Ergebnisse aller in der Studie untersuchten Algorithmen. Allerdings können zu diesem Zweck auch direkt die punktweisen Distanzschätzungen genutzt werden, welche vom vorgestellten PHT/FPHT Detektionssystem generiert werden. Die somit erreichte Fehlerstreuung liegt unterhalb von 1/10 Pixel, mit einer zeitlichen Fehlervariation von weniger als 1/20 Pixel. Diese äußerst hohe Disparitätsgenauigkeit ist vergleichbar mit den Ergebnissen der besten dedizierten Matching-Algorithmen.

Über alle untersuchten Algorithmen hinweg werden die größten Fehler durch im Bild ungenau geschätzte Objektgrenzen sowie durch sogenannte Pixel-locking Effekte von diskreten Matching-Methoden verursacht. Es wird jedoch gezeigt, dass diese Artefakte durch geeignete Gegenmaßnahmen weitestgehend kompensiert werden können.

Die durchgeführten Experimente dienen nicht nur der systematischen Bewertung und dem Vergleich von Algorithmenkonzepten und Komponenten, sondern liefern auch wichtige Referenzwerte für die in der Praxis erzielbare Disparitätsgenauigkeit. Darüber hinaus werden in einer separaten Auswertung die erheblichen Auswirkungen von fehlerbehafteten Kalibrierparametern verdeutlicht. Während Fehler im Schielwinkel der Stereokamera in allen Algorithmen eine erwartet systematische, additive Disparitätsabweichung verursachen, führen Fehler im relativen Nickwinkel zu einem teils signifikanten Anstieg der Fehlerstreuung. Hier sind insbesondere Methoden betroffen, welche das Stereo-Matching direkt auf Intensitätsdaten und mithilfe von sehr kleinen Bildfenstern durchführen. Die gleichzeitige Schätzung der Disparität und des durch die Fehlkalibrierung verursachten vertikalen Bildversatzes kann jedoch Abhilfe schaffen. Nichtsdestotrotz zeigen die Ergebnisse einmal mehr die Notwendigkeit von zuverlässigen Methoden zur Selbstkalibrierung auf.

Im Hinblick auf zukünftige Studien ähnlicher Art bietet sich eine naheliegende Erweiterung durch die Hinzunahme von zusätzlichen Stereoalgorithmen an. Außerdem empfiehlt sich die Nutzung von modernen Hochleistungs-LIDARen als Referenzsensorik. Die Qualität der Referenzdaten kann dadurch sowohl bezüglich Dichte als auch Genauigkeit weiter verbessert werden, was eine noch detaillierte Bewertung und Optimierung der untersuchten Stereoalgorithmik erlaubt.

PUBLICATIONS

Some contributions and results of this work have appeared previously in the following publications:

Peter Pinggera, Uwe Franke, and Rudolf Mester. Highly Accurate Depth Estimation for Objects at Large Distances. In *German Conference on Pattern Recognition (GCPR)*, pages 21–30, 2013. (Cited on pages 83, 86, and 90.)

Peter Pinggera, David Pfeiffer, Uwe Franke, and Rudolf Mester. Know Your Limits: Accuracy of Long Range Stereoscopic Object Measurements in Practice. In *European Conference on Computer Vision (ECCV)*, pages 96–111, 2014. (Cited on pages 83, 98, and 105.)

Peter Pinggera, Uwe Franke, and Rudolf Mester. High-Performance Long Range Obstacle Detection Using Stereo Vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015. (Cited on pages 31, 65, and 67.)

Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and Found: Detecting Small Road Hazards for Self-Driving Vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016. (Cited on pages 31, 57, 64, 65, and 66.)

Sebastian Ramos, Stefan Gehrig, Peter Pinggera, Uwe Franke, and Carsten Rother. Detecting Unexpected Obstacles for Self-Driving Cars: Fusing Deep Learning and Geometric Modeling. In *IEEE Intelligent Vehicles Symposium (IV)*, 2017. (Cited on pages 64, 65, 82, and 119.)

ACRONYMS

ADAS	Advanced Driver Assistance System
BB	Bounding Box
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRF	Conditional Random Field
CStix	Cluster-Stixels
DARPA	Defense Advanced Research Projects Agency
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DEM	Digital Elevation Map
DR	Detection Rate
EM	Expectation-Maximization
FCN	Fully Convolutional Network
$\mathbf{FN_p^i}$	Instance-Level False Negative
FP	False Positive
FPGA	Field-Programmable Gate Array
FPHT	Fast Direct Planar Hypothesis Testing
$\mathbf{FP_p}$	Pixel-Level False Positive
FPR	False Positive Rate
$\mathbf{FPR_p}$	Pixel-Level False Positive Rate
GLRT	Generalized Likelihood Ratio Test
GPU	Graphics Processing Unit
iInt	Instance-Level Intersection
iIoU	Instance-Level Intersection over Union
LDM	Local Differential Matching
LIDAR	Light Detection and Ranging
MAP	Maximum a Posteriori

MLE	Maximum Likelihood Estimate
M-LDM	Multi-Local Differential Matching
MSEG	Joint Matching and Segmentation
PC	Point Compatibility
PDF	Probability Density Function
PHT	Direct Planar Hypothesis Testing
PLC	Pixel Locking Compensation
RADAR	Radio Detection and Ranging
ROC	Receiver-Operator-Characteristic
SGM	Semi-Global Matching
$\mathbf{TP_p^i}$	Instance-Level True Positive
$\mathbf{TP_p}$	Pixel-Level True Positive
TPR	True Positive Rate
$\mathbf{TPR_p}$	Pixel-Level True Positive Rate
TV	Total Variation

Matrices, Vectors and Functions

$\mathbf{M}$	Matrix of arbitrary size
$\mathbf{M}_{(n\times m)}$	Matrix with n rows and m columns
$\mathbf{M}_{i,:}$	ith row of matrix $\mathbf{M}$
$\mathbf{M}_{:,j}$	jth column of matrix $\mathbf{M}$
$\mathbf{M}_{i,j}$	i,jth element of matrix $\mathbf{M}$
$\mathbf{I}$	Identity matrix
$\mathbf{0}$	Zero matrix
$\vec{v}$	Column vector of arbitrary size
$\vec{v}_{(n)} = (v_1, \ldots, v_n)^T$	Column vector of length n
$\vec{0}$	Zero vector
$f(\vec{x}),\ F(\vec{x})$	Scalar-valued function of $\vec{x}$
$\vec{f}(\vec{x}),\ \vec{F}(\vec{x})$	Vector-valued function of $\vec{x}$

Probability Theory and Statistics

$X \sim \mathcal{N}(\mu, \sigma^2)$	Real-valued random variable X, normally distributed with mean μ and variance σ^2
$\vec{X} \sim \mathcal{N}(\vec{\mu}, \mathbf{\Sigma})$	Real-valued random vector $\vec{X}$, normally distributed with mean vector $\vec{\mu}$ and covariance matrix $\mathbf{\Sigma}$
x	Realization of random variable X
$\vec{x}$	Realization of random vector $\vec{X}$
$p(\vec{x})$	Probability Density Function (PDF) of $\vec{X}$
$p(\vec{x};\vec{\theta})$	PDF of $\vec{X}$ parameterized by $\vec{\theta}$
$p(\vec{x}_1\|\vec{x}_2)$	Conditional PDF of $\vec{X}_1$, conditioned on $\vec{X}_2 = \vec{x}_2$
$Pr(A)$	Probability of event A
$Pr(A\|B)$	Conditional probability of event A, conditioned on event B
$\hat{\theta}(\vec{X})$	Estimator for a parameter θ, depending on $\vec{X}$
$\hat{\theta}(\vec{x}),\ \hat{\theta}$	Estimate of θ for an observed realization $\vec{x}$

Hypothesis Testing

$\vec{x}$	Set of observed data samples, i. e. realizations of a random variable X or a random vector $\vec{X}$
$\mathcal{H}_0$	Null hypothesis
$\mathcal{H}_1$	Alternative hypothesis
$p(\vec{x};\mathcal{H}_i)$	PDF of $\vec{X}$ when hypothesis $\mathcal{H}_i$ is true
$p(\vec{x};\vec{\theta},\mathcal{H}_i)$	PDF of $\vec{X}$ parameterized by $\vec{\theta}$ when hypothesis $\mathcal{H}_i$ is true
$Pr(\mathcal{H}_j;\mathcal{H}_i)$	Probability of deciding $\mathcal{H}_j$ when $\mathcal{H}_i$ is true
$Pr_D = Pr(\mathcal{H}_1;\mathcal{H}_1)$	Probability of detection
$Pr_{FA} = Pr(\mathcal{H}_1;\mathcal{H}_0)$	Probability of false alarm
L	Likelihood ratio
L_G	Generalized likelihood ratio
γ	Decision threshold

Point Coordinates

$\vec{X} = (X, Y, Z)^T$	3D point specified in the general world coordinate system
$\tilde{\vec{X}}$	Homogeneous vector representation of $\vec{X}$
$\vec{X}_C = (X_C, Y_C, Z_C)^T$	3D point specified in the camera coordinate system
$\tilde{\vec{X}}_C$	Homogeneous vector representation of $\vec{X}_C$
$\vec{x} = (x, y)^T$	Image point location given in pixels
$\tilde{\vec{x}}$	Homogeneous vector representation of $\vec{x}$
$\tilde{\vec{x}}_s = (x_l, y_l, d, 1)^T$	Extended homogeneous stereo image coordinate vector, including the pixel coordinates (x_l, y_l) in the reference image (left) and the stereo disparity d

Image Data

$\vec{\mathcal{I}}$	Set of images
$I,\ I^t$	Image taken at the current time step t
I^{t-1}	Image taken at the previous time step $t-1$
$I_l,\ I_r$	Left and right image of a stereo pair
$I(\vec{x})$	Image intensity value
$f(\vec{x})$	True intensity value, observed as $I(\vec{x})$

$\alpha(\vec{x})$	Image intensity offset
$\eta(\vec{x})$	Image intensity noise sample, drawn from a zero-mean distribution with variance σ^2
$\vec{\nabla} I(\vec{x}) = \begin{pmatrix} \nabla I_x(\vec{x}) \\ \nabla I_y(\vec{x}) \end{pmatrix}$	Image intensity gradient
W, H	Image dimensions (width, height)
$\vec{x}_c = (x_c, y_c)^T$	Image patch center location
w, h	Image patch dimensions (width, height)
Ω	Image patch support
$\mathcal{D}$	Stereo disparity map
$d = \mathcal{D}(\vec{x})$	Stereo disparity

Camera Parameters

f	Focal length given in meters
f_x	Focal length given in units of pixel width
f_y	Focal length given in units of pixel height
s	Pixel skew
$\vec{x}_0 = (x_0, y_0)^T$	Principal point in pixels
$\vec{C}$	Camera center, origin of the camera coordinate system
$\mathbf{R}$, $\vec{t}$	Position and orientation of the camera coordinate system (rotation matrix and translation vector)
$\mathbf{K}$	Camera calibration matrix
$\mathbf{P}$	Camera projection matrix
B	Baseline length of a stereo camera
$\grave{\mathbf{P}}$	Extended stereo projection matrix

Numerical Optimization

$F(\vec{\theta})$	Cost function, parameterized by $\vec{\theta}$
Θ	Feasible parameter set
$\vec{\theta}$	Parameter vector
$\Delta\vec{\theta}$	Parameter update vector
$\vec{\theta}^L$	Parameter vector, local parameterization
$\Delta\vec{\theta}^L$	Parameter update vector, local parameterization
$\vec{\theta}^*$	Local minimizer of F
$\hat{\vec{\theta}}$	Estimate of $\vec{\theta}^*$

$\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta})$	Parameter update operator
r	Residual, element of the cost function F
$\rho(r)$	Loss function scaling the residual r
ρ', ρ''	First and second derivatives of ρ
$\vec{g}_F$	Cost gradient with respect to $\vec{\theta}$
$\vec{g}$	Cost gradient with respect to $\Delta\vec{\theta}$
$\mathbf{J}_F$	Jacobian matrix of F with respect to $\vec{\theta}$
$\mathbf{J}_{\vec{\oplus}}$	Jacobian matrix of $\vec{\oplus}$ with respect to $\Delta\vec{\theta}$
$\mathbf{J}_{F \circ \vec{\oplus}}$	Jacobian matrix of F with respect to $\Delta\vec{\theta}$
$\mathbf{A}_F$	Hessian matrix of F with respect to $\vec{\theta}$
$\mathbf{A}$	Hessian matrix of F with respect to $\Delta\vec{\theta}$
$\mathbf{A}_{GN}$	Gauss-Newton approximation of $\mathbf{A}$
$\mathbf{A}_{LM}$	Levenberg-Marquardt approximation of $\mathbf{A}$

Object Detection

$\mathcal{H}_f$	Free-space hypothesis, $\mathcal{H}_f \equiv \mathcal{H}_0$
$\mathcal{H}_o$	Obstacle hypothesis, $\mathcal{H}_o \equiv \mathcal{H}_1$
$\vec{\theta}_f, \vec{\theta}_o$	Model parameters for free-space and obstacle hypotheses
$\vec{n} = (n_X, n_Y, n_Z)^T$	Orientation of plane normal
D	Plane normal distance to the coordinate system origin
$\check{\varphi}_f, \check{\varphi}_o$	Plane normal angle constraints for free-space and obstacle hypotheses
a	Disparity slope
b	Disparity offset
$\mathbf{H}$	Homography matrix
$\vec{W}(\vec{x}, \vec{\theta}) = \begin{pmatrix} W_x(\vec{x}, \vec{\theta}) \\ W_y(\vec{x}, \vec{\theta}) \end{pmatrix}$	Warp function, transforming the image point location $\vec{x}$, parameterized by $\vec{\theta}$
$\vec{\odot}(\vec{W}_1, \vec{W}_2)$	Warp composition operator
$\mathbf{J}_{\vec{W}}$	Jacobian matrix of $\vec{W}$ with respect to $\Delta\vec{\theta}$
$\mathbf{J}_{\vec{\odot}}$	Jacobian matrix of $\vec{\odot}$ with respect to $\vec{W}$

Object Tracking

k	Discrete time index
$\vec{\mathcal{X}}_k$	State vector at time step k
$\vec{\omega}_k$	System model noise vector

$\mathbf{Q}_k$	Covariance matrix of $\vec{\omega}_k$
$\mathbf{A}_k$	State transition matrix
$\mathbf{R}_k, \vec{t}_k$	Motion of the ego-vehicle (rotation matrix and translation vector)
$\vec{z}_k$	Measurement vector
$\vec{v}_k$	Measurement noise vector
$h\left(\vec{\mathcal{X}}_k, \vec{v}_k\right)$	Measurement function

Distance Estimation

d^*	True disparity value
$\hat{d}$	Estimated disparity value
$\epsilon_d = \hat{d} - d^*$	Disparity error
$\nabla\epsilon_d = \epsilon_{d,t} - \epsilon_{d,t-1}$	Temporal disparity error variation
Z_C^*	True distance value
$\hat{Z}_C$	Estimated distance value
$\epsilon_Z = Z_C^* - \hat{Z}_C$	Distance error
k	Image segment index
Ω_k	Image segment support
$d_k(\vec{x})$	Disparity of segment model k at $\vec{x}$
$\tilde{d}_k(\vec{x})$	Apparent disparity at $\vec{x}$
$\vec{v} = (v_x, v_y)^T$	Optical flow vector
$\vec{v}_k(\vec{x})$	Optical flow of segment model k at $\vec{x}$
$\tilde{\vec{v}}_k(\vec{x})$	Apparent optical flow at $\vec{x}$
$\vec{s} = (v_x, v_y, d)$	Scene flow vector
$\vec{s}_k(\vec{x})$	Scene flow of segment model k at $\vec{x}$
$i_k(\vec{x})$	Intensity of segment model k at $\vec{x}$
ℓ	Pixel labeling (label configuration)
l	Pixel label (single pixel)
$\vec{\vartheta}$	Optical flow parameter vector
$\vec{\delta}$	Disparity parameter vector
$\vec{\theta}$	Scene flow parameter vector
$\mathcal{C}$	Cliques of a factor graph
V	Clique potential
E	Energy function
S_n	Rousseeuw-Croux scale estimator

CONTENTS

1 INTRODUCTION

CONTENTS

1.1 MOTIVATION AND PROBLEM STATEMENT

1.1.1 *From Driver Assistance Systems to Autonomous Driving*

Over the last decades the annual amount of kilometers traveled by motorized vehicles in Germany has continued to increase steadily, whereas the number of traffic accidents, and most importantly the number of accidents resulting in serious injury or loss of life, has decreased significantly [ADAC, 2016, Statistisches Bundesamt, 2016]. Aside from refined traffic regulations, improved infrastructure and driver training, this positive trend is in large part due to technological advances in driver safety and assistance systems [Lie et al., 2006, Fach and Ockel, 2009]. While passive safety systems such as seat belts and airbags considerably reduce the risk of serious passenger injury in case of an accident, active driver assistance systems aim to prevent accidents from happening in the first place. Various active systems have been established as a de facto standard in modern vehicles, two prime examples being the anti-lock braking system and the electronic stability control system. These standard systems are nowadays being supplemented by more and more Advanced Driver Assistance Systems (ADASs), which are making their way from expensive top-of-the-line models to the bulk of the manufacturers' product ranges. Modern ADASs provide important safety features such as blind spot monitoring, pedestrian recognition and pre-crash braking, but also include convenience functions such as adaptive light control, traffic sign recognition and adaptive cruise control with lane keeping. State-of-the-art cruise control systems may be considered as partially autonomous driving features, being able to accelerate, brake and steer while reacting to the environment within certain constraints. However, these systems

still strictly rely on a human supervisor inside the loop, in order to hand over control in case of unexpected events. Thus, they are still a long way from fully autonomous driving capabilities [SAE, 2016].

Fully autonomous driving technology, with no requirement for human supervision, may be considered as a key enabler for revolutionizing private transport and bringing about several potential benefits, among them the further drastic reduction in traffic accidents, improved traffic flow, reduced parking congestion, better fuel efficiency as well as more efficient use of travel time. Even new mobility solutions and transportation business models such as autonomous car-sharing are conceivable [Lutin et al., 2013, Litman, 2015].

Research towards fully autonomous driving began as early as 1986 with the EUREKA Prometheus project, where substantial commitment by universities as well as industrial research partners resulted in significant advances in this area [Dickmanns et al., 1994, Braess and Reichart, 1995a,b]. In the early 2000s, progress was driven in particular by the Defense Advanced Research Projects Agency (DARPA) autonomous driving challenges, including the 2004 and 2005 *Grand Challenge* [Iagnemma et al., 2006a,b, Thrun, 2006] as well as the 2007 *Urban Challenge* [Urmson, 2008, Montemerlo et al., 2008, Kammel et al., 2008, Miller and Campbell, 2008]. Since then, interest in autonomous driving research and technology has increased dramatically, pushing the state-of-the-art forward as reported in [Guizzo, 2011, Levinson et al., 2011, Bertozzi et al., 2011, Franke et al., 2013, Ziegler et al., 2014] amongst others. However, due to the vast complexity of the problem, combined with the strict legal regulations and safety standards, development is still ongoing and the unconstrained deployment of autonomous vehicles in everyday traffic still seems to be several years away.

1.1.2 *Visual Perception for Intelligent Vehicles*

In order for an autonomous vehicle to navigate through traffic safely and efficiently, it first and foremost needs to be able to perceive and understand its surroundings. Consequently, powerful environment perception systems combining multiple sensor modalities are a critical part of any fully or partially autonomous road vehicle. Besides active sensors such as RADAR and LIDAR, cameras represent a central building block of such systems. While active range sensors offer supreme accuracy in terms of point-wise distance and velocity measurement, they usually suffer from low resolution and high cost. Cameras, on the other hand, are readily available at relatively low cost and can leverage very high image resolution for visual object detection as well as appearance-based semantic reasoning. The use of stereo or multi-camera setups even allows for spatial perception and image-based distance estimation, making such configurations increasingly popular for application in mobile robots in general and autonomous cars in particular.

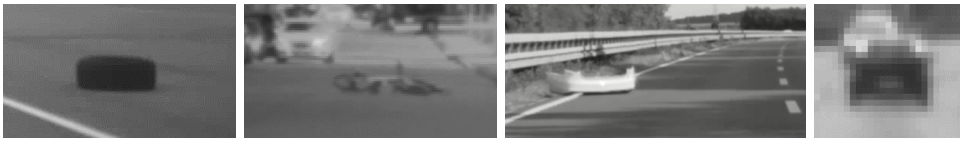


Figure 1.1: The detection and accurate localization of distant and/or small generic obstacles represents a major challenge for perception systems of intelligent vehicles.

1.1.3 *High-Sensitivity Object Detection and High-Accuracy Distance Estimation*

Arguably the most fundamental and crucial task of environment perception systems for autonomous vehicles is the reliable detection and localization of obstacles in order to avoid collisions. Fig. 1.1 illustrates two of the most demanding tasks for visual perception systems within this context:

- The detection of small, yet critical, unexpected obstacles on the road ahead, such as lost cargo or debris.
- The detection and accurate localization of objects at long range, for example the tail end of a traffic jam on a highway.

Note that both tasks are particularly relevant when the ego-vehicle is moving at significant speed. Consequently, the following requirements on visual perception systems arise, which pose a significant challenge to both the sensor hardware as well as the sensor data processing algorithms:

1. *High-sensitivity generic object detection*[1]
   In order to be able to reliably detect generic obstacles even in challenging cases, i. e. objects at long range as well as particularly small and unexpected objects, the perception system has to be sufficiently sensitive. However, at the same time it has to remain robust to real-world conditions, keeping false positive detections at a minimum. The detection system has to be able to handle all types of potential obstacles and must not be limited to a predefined set of object classes.

2. *High-accuracy distance estimation*[2]
   For an autonomous vehicle to react appropriately to detected objects, the location of the potential obstacles has to be determined

1 The terms *object detection* and *obstacle detection* will be used interchangeably throughout this work.

2 Unless otherwise noted, we use the term *accuracy* to describe the combination of both statistical bias (systematic error) as well as statistical variability (random error). *High-accuracy distance estimation* thus implies good trueness (low bias) as well as high precision (low variability) of the distance estimates provided by the system.

as accurately as possible. In particular, the longitudinal distance between the obstacle and the ego-vehicle often represents the most critical parameter for planning emergency braking or other evasive maneuvers. Note that for the estimation of relative object velocities, errors in distance measurements do have a dramatic impact. Unfortunately, compared to active range sensors, camera-based distance estimation is particularly error-prone, requiring highly optimized algorithms to achieve adequate performance.

1.2 CONTRIBUTIONS

The present work directly addresses the requirements and challenges described in the preceding section.

First, a novel method for the visual detection of small, generic obstacles even at long range using stereo cameras is presented. The approach does not make overly restrictive assumptions on the environment and is not limited to predefined sets of semantic object classes. Object detection is based on sound statistical tests using local geometric criteria, which are applicable to both static and moving obstacles and implicitly consider non-flat ground surfaces. The core concept lends itself to massive parallelization and can be implemented efficiently on dedicated hardware. Furthermore, the proposed system supports multiple flexible output representations, ranging from raw 3D obstacle point measurements to compact and generic mid-level obstacle elements to bounding box representations of individual object instances suitable for model-based tracking algorithms.

To allow for a comprehensive evaluation of the detection performance, a dedicated dataset focusing on small and hard-to-detect objects is presented. The dataset is made available to the public to foster further research on this important topic. To the best of the author's knowledge, the proposed system is the first to successfully tackle this specific problem using a standard stereo camera setup and is shown to significantly outperform current general-purpose obstacle detection approaches. Moreover, the presented detection appraches are easily extensible to multi-view camera configurations, providing a promising source of additional performance improvement.

Given that a relevant object has successfully been detected, this work investigates methods for optimizing the stereo-based distance estimation accuracy, particularly at long range. The analysis considers several state-of-the-art stereo algorithm concepts and proposes new approaches for optimizing performance, also taking the trade-off between accuracy and computational complexity into account. A comprehensive statistical evaluation is performed on an extensive dedicated dataset, establishing reference values for the accuracy limits actually achievable in practice. The proposed obstacle detection algorithms are shown to simultaneously

yield highly accurate estimates of object distances, on par with the top-performing dedicated stereo matching algorithms.

1.3 ORGANIZATION OF THE THESIS

This thesis is organized as follows. Chapter 2 provides an introduction to the relevant technical background, including numerical optimization techniques, model selection methods and the fundamentals of stereo vision and multi-view geometry. Additionally, the mathematical terms and notation used in the subsequent chapters are clarified. Chapter 3 describes the proposed high-sensitivity generic object detection system in detail, including the derivation of different algorithmic variants tuned for either flexibility or efficiency. Also, suitable methods for generating different output obstacle representations are presented. The performance of the proposed detection system is evaluated on two challenging datasets, which cover small obstacle occurrences as well as long range detection scenarios. Chapter 4 then provides an analysis of the state-of-the-art in stereo-based distance estimation and proposes various methods for improving long range accuracy. This is followed by an extensive evaluation and comparative study of the presented approaches. Finally, Chapter 5 concludes this work by summarizing the main findings and providing an outlook on future research in this area.

2

TECHNICAL BACKGROUND

CONTENTS

2.1 NON-LINEAR OPTIMIZATION

This section gives a brief overview of selected methods for solving unconstrained as well as constrained non-linear optimization problems which are relevant for the present work. For a more detailed discussion of non-linear optimization in general, the reader is referred to dedicated literature such as [Nocedal and Wright, 1999, Frandsen et al., 2004, Madsen et al., 2004] and references therein.

We consider a scalar-valued cost function F consisting of a sum of residuals r_j scaled by a loss function ρ. The cost function is parameterized by the vector $\vec{\theta} = (\theta_1, \ldots, \theta_n)^T$:

$$F(\vec{\theta}) = \sum_j \rho\left(r_j(\vec{\theta})\right). \tag{2.1}$$

We aim to find a local minimizer $\vec{\theta}^*$ of F within a region of size $\epsilon > 0$ such that

$$F(\vec{\theta}^*) \leq F(\vec{\theta}) \qquad \text{for} \qquad ||\vec{\theta}^* - \vec{\theta}|| < \epsilon. \tag{2.2}$$

A necessary condition for being a local minimizer is for $\vec{\theta}^*$ to be a stationary point, i.e. $\vec{g}_F(\vec{\theta}^*) = \vec{0}$, where $\vec{g}_F$ is the gradient of the cost function with respect to the parameter vector $\vec{\theta}$:

$$\vec{g}_F(\vec{\theta}) = \left(\frac{\partial F}{\partial \theta_1}(\vec{\theta}), \quad \dots \quad , \quad \frac{\partial F}{\partial \theta_n}(\vec{\theta})\right)^T. \tag{2.3}$$

Further, a sufficient condition for $\vec{\theta}^*$ to be a local minimizer is for the Hessian at $\vec{\theta}^*$ to be positive definite, where the elements of the Hessian $\mathbf{A}_F$ represent the second-order derivatives of F, i. e.

$$A_{F_{k,l}}(\vec{\theta}) = \frac{\partial^2 F}{\partial \theta_k \partial \theta_l}(\vec{\theta}). \tag{2.4}$$

Consequently, if $\vec{\theta}^*$ is a stationary point and the corresponding Hessian is positive definite, $\vec{\theta}^*$ is a local minimizer of F [Madsen et al., 2004].

In practice, iterative approaches are typically used to obtain an estimate $\hat{\vec{\theta}}$ of $\vec{\theta}^*$. Starting from a suitable initial parameter vector $\vec{\theta}_0$, in each iteration an update is applied to the current parameter estimate according to

$$\vec{\theta} \leftarrow \vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}) \qquad \text{s.t.} \qquad F\left(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta})\right) < F(\vec{\theta}), \tag{2.5}$$

where $\Delta\vec{\theta} = (\Delta\theta_1, \dots, \Delta\theta_m)^T$ is the update vector and $\vec{\oplus}$ denotes the update operator. Usually, $\vec{\oplus}$ corresponds to a simple additive update with $m = n$, i.e. $\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}) = \vec{\theta} + \Delta\vec{\theta}$. More involved updates sometimes become necessary due to a local reparameterization of the cost function, as is utilized in Sect. 3.2.4.1. Note that in such a case the domains of F and $\vec{\oplus}$, and hence the dimensions of $\vec{\theta}$ and $\Delta\vec{\theta}$, do not necessarily have to be identical. However, the codomain of $\vec{\oplus}$ has to correspond to the domain of F, and $\vec{\oplus}$ has to fulfill the identity relationship $\vec{\oplus}(\vec{\theta}, \vec{0}) = \vec{\theta}$.

The goal of the optimization procedure is for the sequence of steps to converge to $\vec{\theta}^*$. Stopping criteria to assess convergence can be based on the length of the update step or the change in the value of the cost function.

There exist various methods for computing the direction and length of the update vector $\Delta\vec{\theta}$ in each step, resulting in different convergence properties and computational requirements. The following subsections briefly describe the update vector computation strategies relevant for this work.

2.1.1 *Newton's Method*

Newton's method for computing parameter updates can be derived from the condition that $\vec{\theta}^*$ be a stationary point. Denoting the gradient and the

Hessian of the cost function with respect to the parameter update as $\vec{g}$ and $\mathbf{A}$ respectively, we use a Taylor series to describe the expected value of the gradient at $\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta})$ as

$$\begin{aligned}\vec{g}^T\left(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta})\right) = \vec{g}^T\left(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta})\right)\Big|_{\Delta\vec{\theta}=\vec{0}} \\ + \mathbf{A}\left(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta})\right)\Big|_{\Delta\vec{\theta}=\vec{0}} \cdot \Delta\vec{\theta} \\ + O\left(||\Delta\vec{\theta}||^2\right).\end{aligned} \tag{2.6}$$

By truncating the series after the first-order term and evaluating the terms of the right-hand side at $\Delta\vec{\theta} = \vec{0}$, this expression reduces to

$$\vec{g}^T\left(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta})\right) \approx \vec{g}^T(\vec{\theta}) + \mathbf{A}(\vec{\theta})\Delta\vec{\theta}. \tag{2.7}$$

Setting the gradient to zero yields the system

$$-\vec{g}^T(\vec{\theta}) = \mathbf{A}(\vec{\theta})\Delta\vec{\theta}, \tag{2.8}$$

which can then be solved for $\Delta\vec{\theta}$. The result is a descent direction if $\mathbf{A}$ is positive definite. If the initial parameter vector $\vec{\theta}_0$ is sufficiently close to the correct solution and $\mathbf{A}$ is positive definite, Newton's method converges quadratically to $\vec{\theta}^*$. However, if starting farther away from $\vec{\theta}^*$, the algorithm can easily diverge [Frandsen et al., 2004]. Furthermore, $\mathbf{A}$ may not be positive definite or may even be singular so that the system in (2.8) cannot be solved.

Finally, Newton's method always requires calculating the second-order derivatives in $\mathbf{A}$, which is usually analytically cumbersome. Evaluating and inverting $\mathbf{A}$ in each iteration is also computationally expensive for high-dimensional parameter spaces.

2.1.2 *Gauss-Newton Algorithm*

The Gauss-Newton algorithm is a popular and efficient method for solving non-linear least squares systems, but it can also be applied to functions of the form (2.1). The algorithm can be derived from Newton's method by analyzing the components of the gradient $\vec{g}$ and the Hessian $\mathbf{A}$ in (2.6). For the gradient and hence the Jacobian $\mathbf{J}_{F\circ\vec{\oplus}}$ of the cost function with respect to the parameter update we obtain

$$\vec{g}^T\left(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta})\right) = \mathbf{J}_{F\circ\vec{\oplus}}(\vec{\theta}, \Delta\vec{\theta}) = \mathbf{J}_F(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}))\,\mathbf{J}_{\vec{\oplus}}(\Delta\vec{\theta}), \tag{2.9}$$

where $\mathbf{J}_F$ denotes the Jacobian of the cost function with respect to the full parameter vector and $\mathbf{J}_{\vec{\oplus}}$ represents the Jacobian of the update operator

with respect to the parameter update vector[1]. Here, $\mathbf{J}_F$ can be expressed analytically as follows:

$$\begin{aligned}\mathbf{J}_F(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta})) &= \frac{\partial F(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}))}{\partial(\oplus_1(\vec{\theta}, \Delta\vec{\theta}), \ldots, \oplus_n(\vec{\theta}, \Delta\vec{\theta}))} \\ &= \frac{\partial F(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}))}{\partial(\oplus_1(.), \ldots, \oplus_n(.))} \\ &= \frac{\partial \sum_j \rho\left(r_j(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}))\right)}{\partial(\oplus_1(.), \ldots, \oplus_n(.))} \\ &= \sum_j \rho'\left(r_j(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}))\right) \frac{\partial r_j(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}))}{\partial(\oplus_1(.), \ldots, \oplus_n(.))},\end{aligned} \tag{2.10}$$

where ρ' denotes the derivative of the loss function, computed as described in Sect. 2.1.5. The Jacobian of the update operator $\mathbf{J}_{\vec{\oplus}}$ is expressed as

$$\begin{aligned}\mathbf{J}_{\vec{\oplus}}(\Delta\vec{\theta}) &= \frac{\partial(\oplus_1(\vec{\theta}, \Delta\vec{\theta}), \ldots, \oplus_n(\vec{\theta}, \Delta\vec{\theta}))}{\partial(\Delta\theta_1, \ldots, \Delta\theta_m)} \\ &= \frac{\partial(\oplus_1(.), \ldots, \oplus_n(.))}{\partial(\Delta\theta_1, \ldots, \Delta\theta_m)}.\end{aligned} \tag{2.11}$$

The Hessian is obtained by taking the partial derivatives of the cost function gradient with respect to the components of $\Delta\vec{\theta}$. Using the components of $\mathbf{J}_F$ and $\mathbf{J}_{\vec{\oplus}}$ from (2.10) and (2.11) to define the auxiliary variable

$$\mathbf{J}^*(j, \vec{\theta}, \Delta\vec{\theta}) = \frac{\partial r_j(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}))}{\partial(\oplus_1(.), \ldots, \oplus_n(.))} \frac{\partial(\oplus_1(.), \ldots, \oplus_n(.))}{\partial(\Delta\theta_1, \ldots, \Delta\theta_m)}, \tag{2.12}$$

we can write the Hessian as

$$\begin{aligned}&\mathbf{A}(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta})) \\ &\quad= \sum_j \rho''\left(r_j(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}))\right) \cdot \mathbf{J}^{*\,T}(j, \vec{\theta}, \Delta\vec{\theta}) \mathbf{J}^*(j, \vec{\theta}, \Delta\vec{\theta}) \\ &\quad+ \sum_j \rho'\left(r_j(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}))\right) \cdot \frac{\partial\left(\mathbf{J}_1^*(j, \vec{\theta}, \Delta\vec{\theta}), \ldots, \mathbf{J}_m^*(j, \vec{\theta}, \Delta\vec{\theta})\right)}{\partial(\Delta\theta_1, \ldots, \Delta\theta_m)}.\end{aligned} \tag{2.13}$$

The Gauss-Newton algorithm utilizes an approximation of $\mathbf{A}$ by making the assumption that the second-order derivative terms in $\mathbf{A}$ are far smaller than the first-order terms and can therefore be neglected. This assumption holds if the residuals are small or have only little curvature.

1 $\mathbf{J}_{\vec{\oplus}} = \mathbf{I}_{(n \times n)}$ for the case of simple additive parameters updates.

Dropping all second-order terms then yields the approximate Gauss-Newton Hessian

$$\begin{aligned}\mathbf{A}_{GN}&\left(\vec{\oplus}(\vec{\theta},\Delta\vec{\theta})\right)\\&=\sum_j \rho''\left(r_j(\vec{\oplus}(\vec{\theta},\Delta\vec{\theta}))\right)\cdot \mathbf{J}^{*\,T}(j,\vec{\theta},\Delta\vec{\theta})\mathbf{J}^*(j,\vec{\theta},\Delta\vec{\theta}),\end{aligned} \tag{2.14}$$

which is significantly simpler to compute than the full Hessian $\mathbf{A}$. Evaluating $\mathbf{A}_{GN}$ at $\Delta\vec{\theta}=\vec{0}$ yields

$$\begin{aligned}\mathbf{A}_{GN}&\left(\vec{\oplus}(\vec{\theta},\Delta\vec{\theta})\right)\Big|_{\Delta\vec{\theta}=\vec{0}}=\mathbf{A}_{GN}(\vec{\theta})\\&=\sum_j \rho''\left(r_j(\vec{\theta})\right)\cdot \mathbf{J}^{*\,T}(j,\vec{\theta},\vec{0})\mathbf{J}^*(j,\vec{\theta},\vec{0}).\end{aligned} \tag{2.15}$$

The parameter update vector $\Delta\vec{\theta}$ is obtained from (2.8) using $\mathbf{A}_{GN}$. Before performing the actual update step, the length of $\Delta\vec{\theta}$ may additionally be rescaled using line search methods to improve convergence behavior [Nocedal and Wright, 1999]. The Gauss-Newton algorithm generally achieves linear convergence, sometimes increasing to quadratic convergence close to the solution [Madsen et al., 2004].

2.1.3 *Levenberg-Marquardt Algorithm*

The Levenberg-Marquardt algorithms extends the Gauss-Newton algorithm by introducing a damping term to the approximate Hessian:

$$\mathbf{A}_{LM}(\vec{\theta})=\mathbf{A}_{GN}(\vec{\theta})+\gamma\mathbf{D}, \tag{2.16}$$

where $\mathbf{D}$ is a symmetric, positive definite matrix and the magnitude of the damping factor γ is adjusted in each iteration. The damping term influences both the direction and the length of the resulting parameter update vector, making the algorithm behave either closer to the original Gauss-Newton algorithm or to a direct gradient descent approach [Madsen et al., 2004]. Various strategies for formulating and updating the damping term have been proposed, resulting in slight differences regarding update behavior and convergence properties. A common choice is $\mathbf{D}=\operatorname{diag}\left\{\mathbf{A}_{GN}(\vec{\theta})\right\}$ in order to avoid slow convergence behavior in flat regions of the cost function. Generally, the Levenberg-Marquardt algorithm is more robust than the Gauss-Newton algorithm, as it is able to find a solution even when starting far away from the final minimum.

The approach can also be interpreted as a Gauss-Newton algorithm using a trust-region scheme [Nocedal and Wright, 1999].

2.1.4 *Levenberg-Marquardt Algorithm with Convex Constraints*

All algorithms described so far are designed to deal with unconstrained non-linear optimization problems. However, in the present work we also consider tasks that require the solution of non-linear optimization problems with bound constraints. While various methods to tackle constrained optimization problems exist (see [Nocedal and Wright, 1999] for a detailed analysis), here we employ a projected Levenberg-Marquardt approach with convex constraints as presented by Kanzow et al. [2004]. We choose this method due to its efficiency as well as its seamless integration into the general unconstrained optimization framework introduced in the previous sections.

The goal is to find the local minimizer $\vec{\theta}^*$ of the cost function F such that $\vec{\theta}^* \in \Theta$, where Θ represents the feasible set as defined by the given bound constraints. The essential difference to the original Levenberg-Marquardt algorithm is an additional projection step after each parameter update. The projection step ensures valid parameter estimates $\vec{\theta}$ at all times by forcing the updated parameter vector onto the closest point of the feasible set Θ.

While Kanzow et al. [2004] show that this approach can achieve rapid convergence within a local region around the correct solution, the authors propose two extensions to further improve convergence behavior. In case the projected Levenberg-Marquardt step does not result in a sufficient decrease in F, alternatively a line search step in the direction of the obtained projected parameter vector may be applied. If the decrease is still insufficient, a projected line search step in the direction of steepest descent may be taken instead. We refer to [Kanzow et al., 2004] for a detailed derivation and analysis of the complete algorithm.

To solve the constrained non-linear optimization problems considered in this work, we make use of the core projected Levenberg-Marquardt strategy but replace the line search extensions of Kanzow et al. [2004] by a damping factor update scheme as proposed by Madsen et al. [2004]. To assess the quality of each step, the gain ratio as described in the same report is used. The gain ratio evaluates how the actual cost decrease compares to the decrease expected from an unconstrained step based on the predicted cost model. In practical experiments, our modified projected Levenberg-Marquardt algorithm performed on par with the original extended version of Kanzow et al. [2004] while being less complex and more efficient computationally. The full algorithm is detailed in Alg. 2.1.

2.1.5 *Robust Loss Functions*

Considering again the original formulation of the cost function in (2.1), an important factor is the suitable choice of the loss function ρ. Com-

Algorithm 2.1 Constrained Levenberg-Marquardt parameter optimization

Input

- Image data $\vec{\mathcal{I}}$
- Initial parameter vector $\vec{\theta}_0$
- Constraints of feasible set Θ

Output

- Estimate of $\vec{\theta}^*$, i. e. the optimized parameter vector $\hat{\vec{\theta}} \in \Theta$
- Convergence information

Algorithm

function OPTIMIZEPARAMETERS($\vec{\mathcal{I}}$, $\vec{\theta}_0$, Θ)
  converged = false
  $\vec{\theta} = \vec{\theta}_0$
  $\gamma = \gamma_0$
  $\nu = 2$
  while not converged AND iterations<max **do**
    compute error gradient $\vec{g}$
    compute approximate Hessian $\mathbf{A}_{LM}$
    solve $-\vec{g}^T = \mathbf{A}_{LM}\Delta\vec{\theta}$ for update vector $\Delta\vec{\theta}$
    project $\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta})$ onto feasible set: $\vec{\theta}_{prj} = Prj\left(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta})\right)$
    compute gain ratio ξ

    if $\xi > 0$ **then**
      compute step length $||\vec{\theta}_{prj} - \vec{\theta}||$
      compute relative cost change $||\left(F(\vec{\theta}_{prj}) - F(\vec{\theta})\right) / F(\vec{\theta})||$
      update parameter vector: $\vec{\theta} \leftarrow \vec{\theta}_{prj}$
      if step length $< \epsilon_{step}$ OR rel. cost change $< \epsilon_{cost}$ **then**
        converged = true
      end if
      reduce damping factor
      $\gamma = \gamma \cdot \max\left(\frac{1}{3}, 1 - (2\xi - 1)^3\right)$
      $\nu = 2$
    else
      increase damping factor
      $\gamma = \gamma \cdot \nu$
      $\nu = 2 \cdot \nu$
    end if
  end while

  return $\vec{\theta}$, converged
end function

monly, a squared error loss is used, which results in the non-linear least squares problem formulation:

$$F(\vec{\theta}) = \sum_j \rho\left(r_j(\vec{\theta})\right) = \sum_j \left(r_j(\vec{\theta})\right)^2. \tag{2.17}$$

The squared error loss features several desirable properties such as high sensitivity, convexity and continuous derivatives of all orders. The first and second derivatives required for the iterative optimization procedure are simply computed as

$$\rho'(r) = \frac{d\rho(r)}{dr} = 2r, \qquad \rho''(r) = \frac{d^2\rho(r)}{dr^2} = 2, \tag{2.18}$$

where the index j and the parameter vector argument $\vec{\theta}$ have been omitted for brevity.

A major drawback of the squared loss is its susceptibility to outliers in the input data. Even a single outlier in the residuals r_j can cause the resulting parameter estimate $\hat{\vec{\theta}}$ to deviate significantly from the correct solution.

Hence, in the following we shortly describe exemplary alternative loss functions which are robust to outliers and can directly be applied within the optimization framework introduced in the previous sections. For more details on robust loss functions and robust statistics we refer to the respective literature, see for example [Huber and Ronchetti, 2009, Huber, 1964, Hartley and Zisserman, 2004].

2.1.5.1 *Absolute Loss*

The absolute loss, or L1 loss, penalizes residuals by their absolute value

$$\rho(r) = |r|, \tag{2.19}$$

making it much more robust to outliers than the squared loss. While also being continuous and convex, the absolute loss is not continuously differentiable.

To apply the loss in the presented iterative optimization framework, it has to be rewritten as a weighted squared loss:

$$\rho(r) = w^2 r^2, \tag{2.20}$$

where the weight factors $w = \frac{1}{\sqrt{|r|}}$ effectively dampen the influence of large residuals. These damping weights are recomputed in each iteration,

resulting in an iteratively reweighted least squares optimization procedure. The derivatives of the loss function are then computed as

$$\begin{aligned} \rho'(r) &= 2w^2 r, \\ \rho''(r) &= 2w^2. \end{aligned} \tag{2.21}$$

Note that the discontinuity at point $r = 0$ has to be handled with particular care.

2.1.5.2 *Huber Loss*

The Huber loss [Huber, 1964] combines the sensitivity of the squared loss and the robustness of the absolute loss. This is achieved by defining a piecewise function which is convex and continuous with a continuous first derivative. The switch between the two underlying loss models occurs at a given residual threshold value t_h. For small residuals in the range $|r| \leq t_h$ the squared loss is applied, whereas for $|r| > t_h$ the robust absolute loss is used, where

$$w = \frac{\sqrt{2|r|t_h - t_h^2}}{|r|}. \tag{2.22}$$

A smooth approximation can be obtained by the so-called Pseudo-Huber loss [Hartley and Zisserman, 2004]

$$\rho(r) = 2t_h^2 \left(\sqrt{1 + \frac{r^2}{t_h^2}} - 1 \right), \tag{2.23}$$

which features continuous derivatives of all orders.

Fig. 2.1 illustrates the considered loss functions and corresponding residual damping weights, highlighting the beneficial properties of the robust Huber and Pseudo-Huber loss functions.

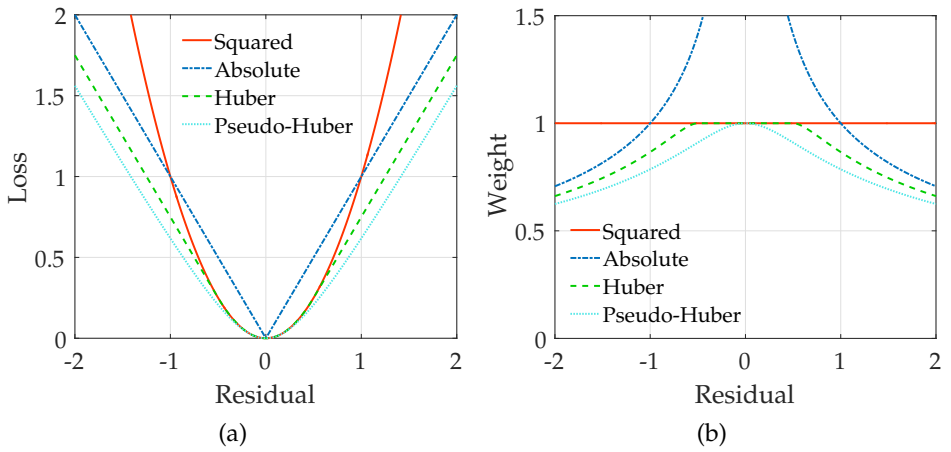


Figure 2.1: Loss functions (a) and respective residual damping weights (b).

2.2 STATISTICAL HYPOTHESIS TESTING FOR MODEL SELECTION

This section introduces the fundamentals of binary statistical hypothesis testing as used for model selection in this work. Further details on statistical decision theory can be found in [Kay, 1998] and references therein.

We consider a set of observed data $\vec{x}$, representing realizations of a random variable or random vector, characterized by one of two possible Probability Density Functions (PDFs). The first PDF $p(\vec{x};\vec{\theta}_0,\mathcal{H}_0)$ is parametrized by the vector $\vec{\theta}_0$ and corresponds to the so-called null hypothesis $\mathcal{H}_0$. The second PDF is parametrized by the vector $\vec{\theta}_1$ and corresponds to the alternative hypothesis $\mathcal{H}_1$. The aim of the hypothesis test is to decide whether the observations $\vec{x}$ originate from $\mathcal{H}_0$ or from $\mathcal{H}_1$.

If the hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ are interpreted to indicate the absence or presence of a signal or an object, then finding an optimal decision criterion can be reformulated as the task of finding an optimal detector. The goal is then to obtain the highest probability of detection $Pr_D = Pr(\mathcal{H}_1;\mathcal{H}_1)$ under a certain probability of false alarms $Pr_{FA} = Pr(\mathcal{H}_1;\mathcal{H}_0)$.

2.2.1 *Simple Hypothesis Testing*

In the case of so-called simple hypotheses, the PDFs under both $\mathcal{H}_0$ and $\mathcal{H}_1$ are fully specified and all parameters are known. The hypotheses are then defined by the respective fixed values of their parameter vectors $\vec{\theta}_0$ and $\vec{\theta}_1$. In such a case, a provably optimal detector can be found by formulating a likelihood ratio test according to the Neyman-Pearson theorem [Neyman and Pearson, 1933]. To maximize Pr_D for a given $Pr_{FA} = \alpha$, the decision for $\mathcal{H}_1$ is taken if

$$L(\vec{x}) = \frac{p(\vec{x};\vec{\theta}_1,\mathcal{H}_1)}{p(\vec{x};\vec{\theta}_0,\mathcal{H}_0)} > \gamma, \tag{2.24}$$

i. e. if the likelihood ratio $L(\vec{x})$ exceeds the threshold γ. The optimal threshold value is computed from

$$Pr_{FA} = \int_{\vec{x}:L(\vec{x})>\gamma} p(\vec{x};\vec{\theta}_0,\mathcal{H}_0)d\vec{x} = \alpha, \tag{2.25}$$

yielding the most powerful among all tests with significance level α.

2.2.2 *Composite Hypothesis Testing*

In practice, the PDFs associated with the competing hypotheses are generally not completely known. The dimensionalities and values of the parameter vectors, even the form of the PDFs, may differ between hy-

potheses. The resulting class of hypothesis testing problems is known as composite hypothesis tests.

2.2.2.1 *Generalized Likelihood Ratio Test*

In the Generalized Likelihood Ratio Test (GLRT) the unknown parameters for each composite hypothesis are replaced by their Maximum Likelihood Estimates (MLEs). Without any further constraints, the GLRT decides for $\mathcal{H}_1$ if

$$L_G(\vec{x}) = \frac{p(\vec{x}; \hat{\vec{\theta}}_1, \mathcal{H}_1)}{p(\vec{x}; \hat{\vec{\theta}}_0, \mathcal{H}_0)} > \gamma, \tag{2.26}$$

with the MLEs of the parameter vectors computed as

$$\hat{\vec{\theta}}_0 = \arg\max_{\vec{\theta}_0} \left(p(\vec{x}; \vec{\theta}_0, \mathcal{H}_0) \right), \tag{2.27}$$

$$\hat{\vec{\theta}}_1 = \arg\max_{\vec{\theta}_1} \left(p(\vec{x}; \vec{\theta}_1, \mathcal{H}_1) \right). \tag{2.28}$$

Given a desired probability of false alarms Pr_{FA}, the threshold parameter γ has to be determined empirically. While this approach provides no general optimality guarantees, it has been shown to perform well in practice [Kay, 1998].

A special case arises if the PDFs under $\mathcal{H}_0$ and $\mathcal{H}_1$ are of the same form $p(\vec{x}; \vec{\theta})$ and the hypothesis test can be formulated as a parameter test of the PDF. Assuming $\vec{\theta} = \left(\vec{\theta}_r, \vec{\theta}_s\right)$ to consist of the parameters to be tested $\vec{\theta}_r$ and a set of nuisance parameters $\vec{\theta}_s$, the test can be written as

$$\mathcal{H}_0 : \quad \vec{\theta} = \left(\vec{\theta}_{r_0}, \vec{\theta}_s\right) \tag{2.29}$$

$$\mathcal{H}_1 : \quad \vec{\theta} \neq \left(\vec{\theta}_{r_0}, \vec{\theta}_s\right). \tag{2.30}$$

The GLRT then decides for $\mathcal{H}_1$ if

$$L_G(\vec{x}) = \frac{p(\vec{x}; \hat{\vec{\theta}}_{r_1}, \hat{\vec{\theta}}_{s_1})}{p(\vec{x}; \vec{\theta}_{r_0}, \hat{\vec{\theta}}_{s_0})} > \gamma. \tag{2.31}$$

Note that the restricted MLE $\hat{\vec{\theta}}_{s_0}$ under $\mathcal{H}_0$ is computed as

$$\hat{\vec{\theta}}_{s_0} = \arg\max_{\vec{\theta}_s} \left(p(\vec{x}; \vec{\theta}_{r_0}, \vec{\theta}_s) \right), \tag{2.32}$$

whereas the unrestricted MLE $\left(\hat{\vec{\theta}}_{r_1}, \hat{\vec{\theta}}_{s_1}\right)$ under $\mathcal{H}_1$ is computed as

$$\left(\hat{\vec{\theta}}_{r_1}, \hat{\vec{\theta}}_{s_1}\right) = \underset{\vec{\theta}_r, \vec{\theta}_s}{\arg\max} \left(p(\vec{x}; \vec{\theta}_r, \vec{\theta}_s)\right). \tag{2.33}$$

In such a special case, the asymptotic detection performance of the GLRT can be determined [Kay, 1998]. For large datasets, it can be shown that the modified test statistic $2 \ln L_G(\vec{x})$ is asymptotically distributed as

$$2 \ln L_G(\vec{x}) \sim \begin{cases} \chi_r^2 & \text{under } \mathcal{H}_0 \\ \chi_r'^2(\lambda) & \text{under } \mathcal{H}_1, \end{cases} \tag{2.34}$$

where χ_r^2 represents a chi-squared distribution with r degrees of freedom and $\chi_r'^2(\lambda)$ denotes a noncentral chi-squared distribution with noncentrality parameter λ. The parameter r is given by the dimensionality of the tested parameter vector $\vec{\theta}_r$. The distribution under $\mathcal{H}_0$ does not depend on any unknown parameters, which allows for the computation of a suitable decision threshold γ in order to maintain a certain probability of false alarms Pr_{FA}.

2.3 STEREO VISION

This section introduces the basic concepts of computational stereo vision, starting with the essential pinhole camera model, the fundamentals of two-view epipolar geometry, image rectification and the so-called standard stereo configuration. Subsequently, the central task of stereo vision, the correspondence problem, is introduced, the solution of which represents the critical prerequisite for successful 3D reconstruction of observed points. Finally, an overview of possible error sources in the various stages of the stereo vision process is given and the respective impact on the present work is discussed.

For a detailed introduction to camera models, projective geometry and multi-view geometry, the reader is referred to the respective literature, in particular [Faugeras and Luong, 2001, Forsyth and Ponce, 2002, Hartley and Zisserman, 2004].

2.3.1 *The Pinhole Camera Model*

The camera model used throughout this work is based on the ideal pinhole camera, which provides a mapping of points from Euclidean 3-space $\mathbb{R}^3$ to Euclidean 2-space $\mathbb{R}^2$ by central projection. The geometric model of the pinhole camera is illustrated in Fig. 2.2. The camera center $\vec{C}$ represents the center of projection and sits at the origin of the camera coordinate system, which we define as a left-handed three-dimensional Cartesian coordinate system. The line perpendicular to the image plane passing through the camera center is called the principal axis, intersecting the image plane at the principal point $\vec{x}_0$. The image coordinate system is defined as a two-dimensional Cartesian coordinate system in the image plane, with its origin located at the principal point. The principal point can also be expressed in camera coordinates as $(0, 0, f)^T$, where f denotes the focal length of the camera. A 3D point $\vec{X}_C = (X_C, Y_C, Z_C)^T$ given in the camera coordinate system is mapped onto the image where the ray connecting $\vec{X}_C$ and $\vec{C}$ intersects the image plane $Z_C = f$. This central projection mapping can be compactly written as

$$\tilde{\vec{x}}_P = \mathbf{P}\tilde{\vec{X}}_C, \tag{2.35}$$

where the 3D point is represented by the homogeneous 4-vector $\tilde{\vec{X}}_C = (X_C, Y_C, Z_C, 1)^T$ and the corresponding image point is represented by the homogeneous 3-vector $\tilde{\vec{x}}_P = (x_P, y_P, 1)^T$. The matrix $\mathbf{P}$ is the 3×4 camera projection matrix of the ideal pinhole camera

$$\mathbf{P} = diag\,(f, f, 1)\left(\mathbf{I}_{(3\times3)} \,|\, \vec{0}_{(3)}\right). \tag{2.36}$$

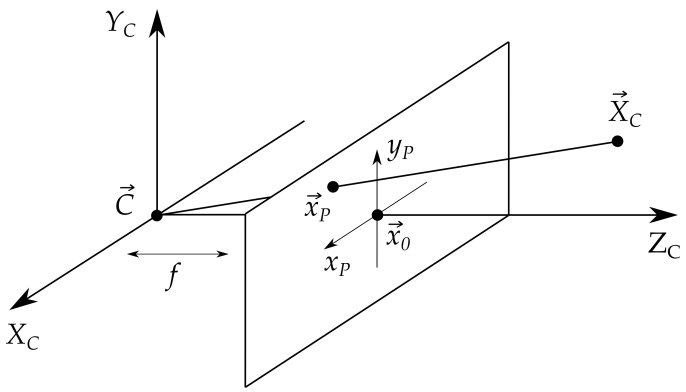


Figure 2.2: The basic pinhole camera model with camera center $\vec{C}$, principal point $\vec{x}_0$ and focal length f.

A more general model extending the ideal pinhole camera is provided by the finite projective camera [Hartley and Zisserman, 2004] with the camera projection matrix

$$\mathbf{P} = \mathbf{K}\left(\mathbf{R} \,|\, \vec{t}\right), \tag{2.37}$$

which includes the extrinsic parameters $\mathbf{R}$ and $\vec{t}$ and the intrinsic parameters in the form of the camera calibration matrix $\mathbf{K}$.

2.3.1.1 *Extrinsic Parameters*

Commonly, an observed 3D point $\vec{X}$ is given in the world coordinate system, which is different from the camera coordinate system in general. The two coordinate systems are related by a rotation and a translation, defined by the 3×3 rotation matrix $\mathbf{R}$ and the translation 3-vector $\vec{t}$. Points are transformed from the world coordinate system to the camera coordinate system by first applying the rotation $\mathbf{R}$, followed by the translation $\vec{t}$:

$$\vec{X}_C = \mathbf{R}\vec{X} + \vec{t}. \tag{2.38}$$

2.3.1.2 *Intrinsic Parameters*

The intrinsic parameters of a practical finite projective camera are represented by the camera calibration matrix $\mathbf{K}$, which is a 3×3 matrix with five degrees of freedom:

$$\mathbf{K} = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.39}$$

The parameters $f_x = f m_x$ and $f_y = f m_y$ specify the focal length in terms of pixel dimensions, with m_x and m_y representing the reciprocal pixel width and height, respectively. The pixel coordinates of the principal

point are defined by $\vec{x}_0 = (x_0, y_0)^T$. Finally, the parameter s represents a potential pixel skew, caused by a non-perpendicular angle between the x- and y-axes of the camera's pixel array. In most practical applications the skew can be considered to be zero.

2.3.1.3 *Lens Correction*

The ideal pinhole camera offers a simple mathematical formulation for the projection of object points to images. However, this ideal model can only provide an approximation of real camera projection, as the lenses employed in practice always introduce some form of distortion or aberration. A widely used formula for modeling lens distortion effects is

$$\vec{x}_D = \vec{x}_P + \Delta\vec{x}_R + \Delta\vec{x}_T, \tag{2.40}$$

where the image coordinates of the ideal pinhole projection are denoted as $\vec{x}_P = (x_P, y_P)^T$ and the distorted coordinates as $\vec{x}_D = (x_D, y_D)^T$. The radial lens distortion component $\Delta\vec{x}_R$ represents a symmetric radial displacement of points in the image plane [McGlone et al., 2004] and can be approximated by

$$\Delta\vec{x}_R = \begin{pmatrix} \Delta x_R \\ \Delta y_R \end{pmatrix} = \left(k_1 r^2 + k_2 r^4 + \ldots\right) \begin{pmatrix} x_P \\ y_P \end{pmatrix}, \tag{2.41}$$

where k_1 and k_2 denote the radial distortion coefficients and $r = \sqrt{x_P^2 + y_P^2}$ is the radial distance from the principal axis.

The decentering distortion component $\Delta\vec{x}_T$ is caused by the fact that the centers of curvature of compound lenses may not be perfectly collinear [Brown, 1966, 1971]. This results in both radial and tangential distortion components, commonly expressed as

$$\Delta\vec{x}_T = \begin{pmatrix} \Delta x_T \\ \Delta y_T \end{pmatrix} = \begin{pmatrix} 2k_3 x_P y_P + k_4(r^2 + 2x_P^2) \\ k_3(r^2 + 2y_P^2) + k_4(r^2 + 2k_4 x_P y_P) \end{pmatrix}, \tag{2.42}$$

with coefficients k_3 and k_4. Combining the intrinsic parameters of the finite projective camera (2.39) with the described lens distortion coefficients yields the projection model

$$\vec{x}_d = \begin{pmatrix} x_d \\ y_d \end{pmatrix} = \begin{pmatrix} m_x x_D + s y_D \\ m_y y_D \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}, \tag{2.43}$$

which yields the final pixel coordinates $\vec{x}_d$ for an observed point $\vec{X}_C$. This formulation provides a direct mapping between the ideal central projection and the distorted image captured by a real camera. If the distortion coefficients are known, it is possible to perform so-called lens correction, hence removing distortion artifacts and reconstructing a virtually

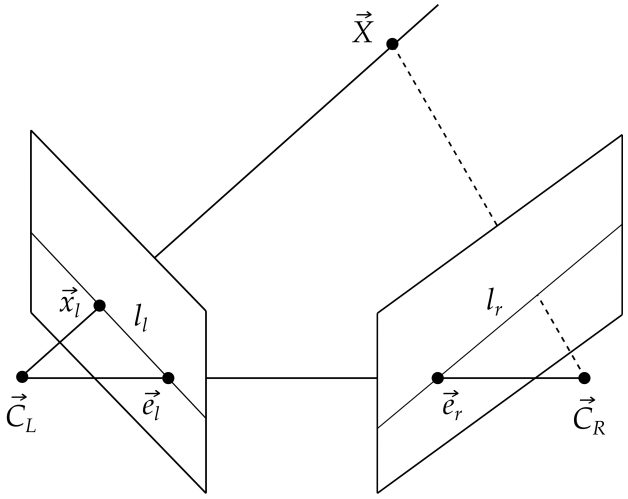


Figure 2.3: Epipolar geometry of a stereo camera setup, with the left and right camera centers $\vec{C}_L$ and $\vec{C}_R$, the epipoles $\vec{e}_l$ and $\vec{e}_r$ and the epipolar lines l_l and l_r.

undistorted image. This allows for the application of the finite projective camera model in subsequent processing steps.

In practice, the distortion coefficients in (2.41) and (2.42) as well as the intrinsic parameters of the calibration matrix (2.39) are estimated in an offline calibration procedure using known calibration targets [Heikkilä and Silven, 1997, Zhang, 2000, Bouguet, 2017].

2.3.2 *Epipolar Geometry*

Stereo vision for 3D perception is based on the use of two vertically aligned[2] cameras with largely overlapping fields of view. This camera configuration is characterized by the so-called epipolar geometry, illustrated in Fig. 2.3 [Faugeras and Luong, 2001, Hartley and Zisserman, 2004]. Assuming two lens-corrected finite projective cameras, a world point $\vec{X}$ is projected into the left and right camera images at pixel coordinates $\vec{x}_l$ and $\vec{x}_r$, respectively. The plane defined by the camera centers $\vec{C}_L$ and $\vec{C}_R$ and the world point $\vec{X}$ is called the epipolar plane. Projecting the epipolar plane into each image yields the epipolar lines l_l and l_r. The baseline connects the two camera centers and intersects the image planes at the epipoles $\vec{e}_l$ and $\vec{e}_r$. Since the projection $\vec{x}_l$ of any point $\vec{X}$ in the left image lies on the epipolar plane, its corresponding point $\vec{x}_r$ in the right image must lie on the epipolar line l_r, and vice versa. This property is known as the epipolar constraint.

In order to reconstruct 3D points from stereo imagery via triangulation, it is essential to first determine point correspondences across the two images. Due to the epipolar constraint, the complexity of the corre-

2 Some applications also use custom setups such as vertically stacked cameras or multi-view rigs.

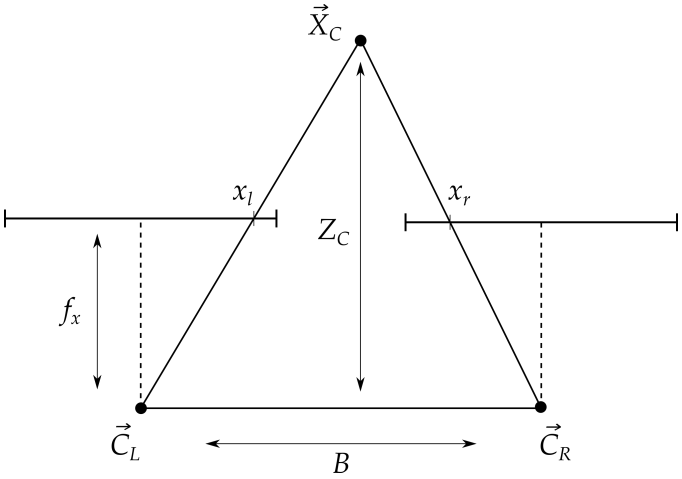


Figure 2.4: Standard stereo configuration.

spondence search is reduced significantly, as the search space is reduced from full images to epipolar lines only.

2.3.2.1 *The Standard Stereo Configuration*

To further reduce the complexity of the correspondence search as well as the triangulation procedure, the camera setup is commonly transformed into the so-called standard stereo configuration as shown in Fig. 2.4. In this configuration, the image coordinate systems of both cameras are perfectly aligned, sharing a single image plane. The baseline separating the camera centers is oriented parallel to the image coordinate x-axis, with the baseline length denoted as B. All intrinsic parameters are shared between both cameras. Epipolar lines are then equivalent to image rows, reducing the correspondence search to a 1D search along corresponding rows. The left camera is defined as the reference camera, with its camera center specifying the origin of the joint stereo camera coordinate system.

RECTIFICATION For the stereo setup to be transformed into its standard configuration, the intrinsic parameters as well as the relative extrinsic parameters of both cameras have to be known. All required parameters can be obtained from offline calibration procedures with known calibration targets [Tsai, 1987, Heikkilä and Silven, 1997, Zhang, 2000, Bouguet, 2017]. In an online rectification process, the original camera images are then warped onto the new image planes satisfying the properties of the standard configuration [Fusiello et al., 2000].

STEREO DISPARITY Given a pair of rectified stereo images and a point $\vec{X}_C = (X_C, Y_C, Z_C)$ in the stereo camera coordinate system, the corresponding projections in the left and right images are denoted as $\vec{x}_l = (x_l, y_l)^T$ and $\vec{x}_r = (x_r, y_r)^T$, with $y_l = y_r$ due to the epipolar constraint. The horizontal displacement, i.e. the difference in x-coordinates, is called

stereo disparity d. It is related to the point distance or depth Z_C via similar triangles:

$$d = x_l - x_r = \frac{f_x B}{Z_C}. \tag{2.44}$$

Using an extended stereo projection matrix $\grave{\mathbf{P}}$ as in [Rabe, 2011]

$$\grave{\mathbf{P}} = \begin{pmatrix} f_x & 0 & x_0 & 0 \\ 0 & f_y & y_0 & 0 \\ 0 & 0 & 0 & f_x B \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \vec{t} \\ \vec{0}_{(3)}{}^T & 1 \end{pmatrix}, \tag{2.45}$$

the projection of a world point $\tilde{\vec{X}} = (X, Y, Z, 1)$ into the stereo camera images can simply be written as the mapping

$$\tilde{\vec{x}}_s = \grave{\mathbf{P}} \tilde{\vec{X}}, \tag{2.46}$$

where $\tilde{\vec{x}}_s$ denotes the extended homogeneous image coordinate vector $\tilde{\vec{x}}_s = (x_l, y_l, d, 1)^T$.

2.3.3 *3D Reconstruction*

In general stereo configurations, the reconstruction of a 3D point from two corresponding image points is performed by calculating the intersection of the back-projected rays passing through the respective camera centers and image points. However, in practice image coordinates as well as camera parameters are only known approximately, and the triangulation task effectively turns into an optimization problem [Hartley and Sturm, 1997, Trucco and Verri, 1998]. In case of the standard stereo configuration, a 3D point can be obtained directly by inverting the projection formula (2.46):

$$\tilde{\vec{X}} = \grave{\mathbf{P}}^{-1} \tilde{\vec{x}}_s. \tag{2.47}$$

2.3.4 *The Correspondence Problem*

The central task of stereo vision is the solution of the correspondence problem or stereo matching problem, i.e. finding corresponding points in stereo imagery. Here we focus on the standard stereo configuration, where the disparity values resulting from correspondences in rectified images allow for the direct reconstruction of 3D scene points via (2.47). A massive amount of research has been published on this topic in the past decades, constantly advancing the state-of-the-art. In particular, progress has been driven by public performance evaluation benchmarks such as

Figure 2.5: Example of a dense disparity map computed via the Semi-Global Matching (SGM) algorithm [Hirschmüller, 2008, Gehrig et al., 2015]. The pixel-wise disparity results are visualized as a color-coded overlay, where green represents small disparity (large distance) and red represents large disparity (small distance).

the *Middlebury* [Scharstein and Szeliski, 2002, Scharstein et al., 2014] and *KITTI* [Geiger et al., 2012, Menze and Geiger, 2015] benchmarks.

Depending on the application scenario, the correspondence search is performed either for a selected sparse set of image points only, or for all image pixels yielding a dense or at least semi-dense correspondence map, the disparity map (Fig. 2.5). The majority of traditional dense stereo matching algorithms can be cast into the general taxonomy proposed by Scharstein and Szeliski [2002], including either all or a subset of the processing steps

- matching cost computation,
- cost aggregation,
- disparity computation,
- disparity refinement.

In the first step, pixel-wise matching costs between all potentially corresponding points in the left and right images are computed based on a given similarity measure. While some similarity measures are derived from statistical models of the image content, others are simply designed with the goal of finding maximally discriminative and robust descriptors of image points. Common measures include squared or absolute intensity differences, cross-correlation-based measures, ordinal measures or differences of intensity derivative signatures. Recently, even the applicability of similarity measures obtained from a supervised learning approach based on Convolutional Neural Networks (CNNs) was demonstrated [Žbontar and LeCun, 2016].

Next, pixel-wise matching costs are aggregated over small local image regions, as similarity measures computed on single pixels are in general too ambiguous for reliable matching. In fact, most similarity measures

implicitly perform a type of local aggregation, as they already make use of a local neighborhood for cost computation.

In the third step, optimal correspondences are selected based on the determined matching costs and a given optimality criterion. Here a distinction is made between so-called local and global algorithms. Local algorithms follow the simple notion that, out of a certain set of correspondence candidates, the one with the highest similarity score should be selected. Consequently, this approach is very susceptible to ambiguities and outliers in the matching costs. In contrast, global algorithms formulate the correspondence selection as an optimization problem, taking the global image context into account. In this way, additional constraints such as scene priors favoring smooth or piece-wise smooth disparity maps can be exploited.

While some algorithms directly treat stereo disparity as a continuous variable, many are set in a discrete framework, which considers disparity on a pixel-discrete grid. For such discrete settings, a final refinement step can be employed to obtain sub-pixel accurate results. Commonly, this is achieved by fractional sampling of the disparity space and/or a curve fit to the computed matching cost volume.

Not all modern stereo matching algorithms strictly conform to the taxonomy described above. For example, a number of current methods combine image segmentation and segment-wise model parameter estimation to achieve state-of-the-art performance [Vogel et al., 2015, Yamaguchi et al., 2014]. Recently, machine learning approaches have successfully been applied to adaptively tune the parameters of disparity optimization algorithms [Seki and Pollefeys, 2017] or even for direct inference of disparity maps via CNNs trained end-to-end [Mayer et al., 2016, Kendall et al., 2017].

Further details on the stereo correspondence problem and an extensive overview of dense matching algorithms can be found in [Szeliski, 2010] and references therein, as well as in the benchmark rankings of [Scharstein et al., 2014, Geiger et al., 2012, Menze and Geiger, 2015]. Chapter 4 of the present work focuses on the specific problem of designing algorithms which yield highly accurate correspondence results in the sub-pixel range, including further discussion of related research.

2.3.5 *Sources of Error*

In all stages of the stereo processing pipeline, several sources for introducing errors and inaccuracies exist, with some errors having a more severe impact on the final 3D reconstruction of points than others. This section is intended to provide an overview of the most common sources of error, along with their respective relevance for the accuracy of reconstruction results.

2.3.5.1 *Calibration*

Accurate camera calibration arguably represents the most critical prerequisite for all further processing steps, as errors in the estimated intrinsic and extrinsic parameters have a direct and severe impact on both the correspondence search as well as 3D reconstruction results.

Regarding intrinsic and lens distortion parameters, in some cases the applied camera model might simply not be able to fully capture the actual camera characteristics, a problem which may be overcome by selecting a more suitable model. However, even for suitable camera models, errors can arise if the set of selected calibration images is not sufficient to constrain the parameter estimation problem, such that the applied optimization procedure is not able to find a high-quality solution. This can also result in the parameters correctly capturing the camera properties only in certain image areas, but not in the full image.

Considering extrinsic camera calibration, a major challenge is the fact that the actual parameters of the camera setup can easily deteriorate during operation due to adverse environmental conditions such as vibrations or temperature variations. These effects require frequent re-calibration or the use of online self-calibration algorithms, see e. g. [Dang et al., 2009].

Inaccurate camera parameter estimates can lead to image rows not exactly fulfilling the epipolar constraint after rectification, either in the full images or in image parts. In practice, this is most often caused by changes in the relative pitch or roll angles of the two cameras over time. If the assumption of image rows coinciding with epipolar lines is violated, matching errors become inevitable. However, such errors may be mitigated by the use of appropriate matching algorithms, e. g. by jointly estimating disparity and vertical offsets as described in Sect. 4.3.3 or by using dedicated image filters and similarity measures as shown in [Hirschmüller and Gehrig, 2009].

Even if correct image correspondences can be found, inaccurate camera parameter estimates directly lead to errors in the 3D reconstruction. In [Zhao and Nandhakumar, 1996], the effects of errors in various camera parameters on point distances computed via (2.47) is analyzed analytically. Not considering potential errors in the baseline, the relative yaw angle between cameras is found to be most critical, followed by pitch and roll angles. A corresponding experimental analysis is provided in [Nedevschi et al., 2003]. Here a general stereo setup is assumed, where triangulation is performed by computing the point closest to the back-projected rays from both cameras [Trucco and Verri, 1998]. The results confirm the intuitive notion that the parameters most critical for accurate distance computation are the horizontal coordinates of the principal points, the baseline length, as well as the relative yaw and pitch angles.

2.3.5.2 *Correspondence*

Despite its perceived conceptual simplicity, the search for stereo correspondences represents a hard problem which holds a vast amount of potential error sources. While the state-of-the-art in stereo matching algorithms has made significant advances in recent years, challenging scenes including large texture-less image areas, occlusions, repetitive patterns, small objects or transparent/semi-transparent materials still present serious difficulties [Scharstein and Szeliski, 2002, Scharstein et al., 2014, Geiger et al., 2012, Menze and Geiger, 2015].

Moreover, inaccuracies or errors are often due to inappropriate model assumptions applied by matching algorithms. For example, the use of inadequate, non-robust similarity measures can result in serious errors if the imaging characteristics of the cameras do not fulfill the underlying assumptions. In global methods, thin structures and small objects tend to get over-smoothed due to the applied scene priors and global smoothness constraints. In local methods on the other hand, image patches used for cost aggregation are sometimes not able to correctly represent the local image context. Matching algorithms based on image segmentation and segment-wise estimation of parametric disparity models yield erroneous disparity maps if the segmentation or the estimated model parameters are inaccurate.

A different type of error arises if disparity values are considered as discrete integer-valued displacements only, corresponding to the pixel resolution of the camera sensor. The resulting quantization errors directly lead to erroneous or at least ambiguous 3D coordinates, as any reconstructed point is essentially represented by a whole region in 3D space. The characteristics of reconstruction errors resulting from disparity quantization have been analyzed in detail by several authors including Blostein and Huang [1987], Matthies and Shafer [1987], Rodriguez and Aggarwal [1990], Chang et al. [1994], Fooladgar et al. [2013] and Freundlich et al. [2015].

To avoid the implications of disparity quantization errors, it is essential to obtain sub-pixel accurate matching results. Even if disparity computation is implicitly restricted to discrete disparity space due to details of the underlying matching algorithm, sub-pixel values can be obtained by appropriate pre- or post-processing steps, such as fractional disparity sampling or matching cost interpolation. However, interpolated sub-pixel disparities can still contain artifacts such as the so-called pixel-locking effect, where the disparity distribution is biased towards integer values [Shimizu and Okutomi, 2001, Nehab et al., 2005, Haller and Nedevschi, 2012].

Last but not least, adverse environmental conditions such as heavy rain, snowfall, fog or low light as well as wet or dirty lenses significantly reduce the quality of the matching results and the computed disparity maps in practice.

2.3.5.3 *Reconstruction*

Leaving aside gross errors caused by imperfect calibration, quantization artifacts or incorrect correspondences, general models for reconstruction uncertainty are usually based on the assumption of normally distributed noise on the image coordinates of corresponding points in the stereo images. While the properties of the reconstruction error distribution can then be specified in all spatial dimensions [Zhang and Boult, 2011], errors in distance far outweigh errors in the other directions [Blostein and Huang, 1987]. The Cramer-Rao lower bound on the distance uncertainty can be derived as shown in [Yang et al., 2010], assuming noisy image coordinates but otherwise perfect camera parameters. Notably, the bound is lowest for points at the image center and increases towards the image borders.

Considering a standard stereo configuration, the assumed noise model on the image coordinates reduces to the horizontal direction only. In terms of disparity measurements this can be written as

$$\hat{d} = d^* + \epsilon_d, \tag{2.48}$$

where d^* denotes the true disparity, $\hat{d}$ denotes the measured value, and the error term ϵ_d represents samples from a symmetric distribution with zero mean and variance σ_d^2. Following (2.44), the corresponding distance value is computed as

$$\hat{Z}_C = \frac{f_x B}{\hat{d}} = \frac{f_x B}{d^* + \epsilon_d}. \tag{2.49}$$

For any given ϵ_d, the resulting distance error ϵ_Z is then

$$\epsilon_Z = Z_C^* - \hat{Z}_C = \frac{f_x B}{d^*} - \frac{f_x B}{d^* + \epsilon_d} = \frac{f_x B}{d^*} \frac{\epsilon_d}{d^* + \epsilon_d} = \frac{{Z_C^*}^2 \epsilon_d}{f_x B + Z_C^* \epsilon_d}. \tag{2.50}$$

It becomes apparent from (2.50) that the distance error ϵ_Z resulting from a constant disparity error ϵ_d increases non-linearly with increasing absolute distance. Fig. 2.6 illustrates this relationship on a set of exemplary disparity error values.

Under the assumption of normally distributed disparity errors, the PDF of the resulting distance estimates $\hat{Z}_C$ is given as derived in [Sibley et al., 2007]:

$$p(\hat{z}_C) = \frac{f_x B}{\sqrt{2\pi}\sigma_d \hat{Z}_C^2} \exp\left(-\frac{\left(\frac{f_x B}{\hat{Z}_C} - d^*\right)^2}{2\sigma_d^2} \right). \tag{2.51}$$

Fig. 2.7 depicts the distance PDFs resulting from an exemplary set of disparity error magnitudes σ_d, for a given true disparity d^* and

corresponding Z_C^*. As pointed out previously by Sibley et al. [2007] and Rabe [2011], the PDF is clearly asymmetric and biased towards larger distances, with the magnitude of the bias increasing with the disparity error and with absolute distance.

Considering the results obtained in (2.50) and (2.51), the implications of which are illustrated in Fig. 2.6 and Fig. 2.7, it is evident that the accuracy of disparity results in the sub-pixel range is a most critical prerequisite for long range distance estimation using stereo cameras.

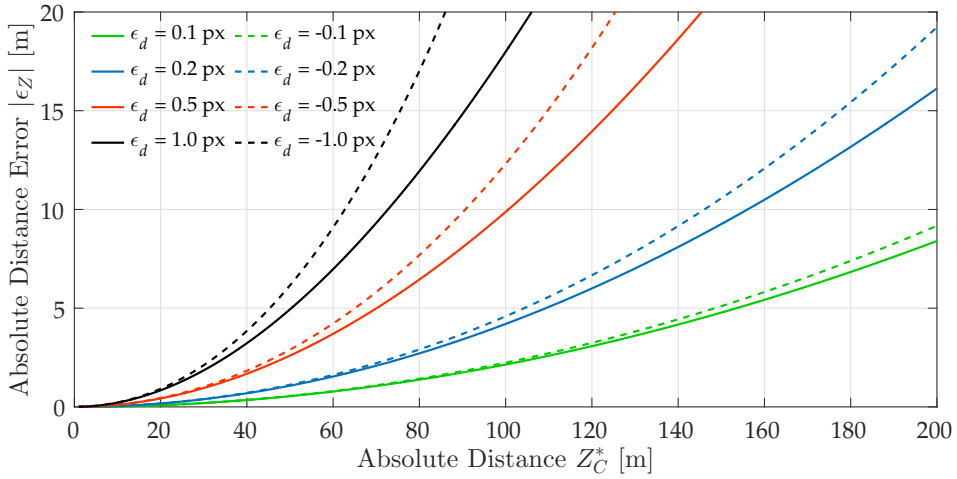


Figure 2.6: Absolute distance errors increase non-linearly for a given set of stereo disparity errors, shown for camera parameters $f_x = 1200$ px and $B = 0.38$ m.

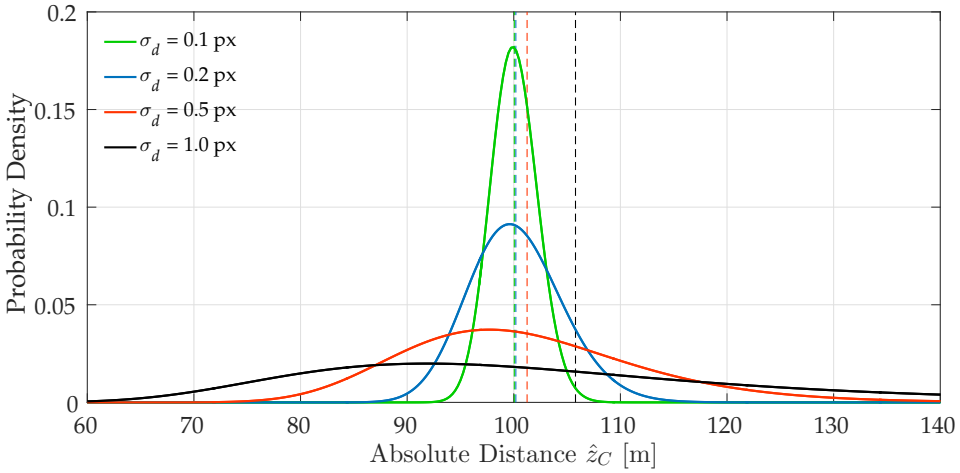


Figure 2.7: Distance PDFs $p(\hat{z}_C)$ for a given set of disparity error standard deviation magnitudes, with the true distance $Z_C^* = 100$ m and camera parameters $f_x = 1200$ px and $B = 0.38$ m. Dashed lines represent the expected values of the respective PDFs, illustrating the distance bias resulting from erroneous disparity values.

3 OBJECT DETECTION

CONTENTS

Parts of this chapter have appeared previously in [Pinggera et al., 2015] and [Pinggera et al., 2016].

3.1 RELATED WORK

3.1.1 *Overview*

Many generic obstacle and object detection approaches for intelligent ground vehicles are based on geometric criteria and employ a so-called

flat-world-assumption, modeling free-space or ground as a single planar surface and characterizing objects by their height-over-ground [Zhang et al., 1997, Lourakis and Orphanoudakis, 1998, Nedevschi et al., 2004b, Bichsel and Borges, 2016]. Geometric deviations from the reference plane can be estimated either from a precomputed point cloud [Nedevschi et al., 2004b, Bichsel and Borges, 2016], directly from image data [Sawhney, 1994], or via mode extraction from a v-disparity histogram [Kramm and Bensrhair, 2012]. However, the resulting detection performance strongly depends on the accuracy of the ground plane parameters as well as the validity of such a simple model. Consequently, more sophisticated ground profile models have been introduced, from piece-wise planar longitudinal profiles [Labayrade et al., 2002] to clothoids [Nedevschi et al., 2004a] and splines [Wedel et al., 2009a]. Also, parameter-free ground profile models have been investigated using multiple filter steps and adaptive thresholding in the v-disparity domain [Harakeh et al., 2015].

The survey in [Bernini et al., 2014] presents an overview of several stereo-based obstacle detection approaches that have proven to perform very well in practice. The considered methods span a range of different ground profile models and object representation categories, including the so-called *Stixel World* [Badino et al., 2009, Pfeiffer and Franke, 2011], Digital Elevation Maps (DEMs) [Oniga and Nedevschi, 2010] and geometric point clusters [Manduchi et al., 2005, Broggi et al., 2011]. The approach of Oniga and Nedevschi [2010] produces a dense scene representation distinguishing free-space and various types of objects by using a DEM in combination with a quadratic ground model. The point cluster method of Manduchi et al. [2005] and the Stixel algorithm of Pfeiffer and Franke [2011] will be described in more detail in Sect. 3.1.2 and Sect. 3.1.3. Notably, all of these methods rely on precomputed stereo disparity maps.

The above methods are designed for robust generic object detection based on different types of stereo-based geometric criteria and work best in close range to medium range applications. Detection performance and object localization accuracy drop quickly with increasing distance. By using custom sensor configurations, such as the trinocular large-baseline tele-stereo setup shown in [Williamson and Thorpe, 1999], impressive performance boosts can be achieved. However, such dedicated hardware setups are often bulky, expensive and reduce the versatility of a given sensor hardware configuration for use in multiple application scenarios. In contrast, the present work focuses on approaches which are applicable to general-purpose stereo cameras and do not strictly require custom sensor hardware. Nevertheless, in Sect. 3.2.6 a straightforward extension of the proposed methods to multi-camera setups is described. A corresponding increase in performance can be expected when utilizing a trinocular setup as in [Williamson and Thorpe, 1999].

A different line of work on generic object detection utilizes appearance cues in addition to geometric criteria to improve detection performance. In [Hadsell et al., 2008], a deep belief network is employed to distinguish

objects from traversable regions for off-road driving. Here, a stereo-based detection algorithm acts as a supervisor for collecting training samples in the short range, while the trained appearance-based classifier yields predictions for the long range. A similar idea is pursued in [Lu et al., 2015], however, instead of actually training a classifier, spectral clustering of superpixels is applied to perform the long range prediction. The reported results appear promising but rather coarse for reliable long range object reasoning. In [Creusot and Munawar, 2015] the detection of obstacles on the road is mapped to the task of appearance-based anomaly detection and tackled via a Restricted Boltzmann Machine neural network. Patches which deviate from the learned road appearance model are considered a potential hazard. However, due to the lack of geometric information, the method tends to trigger on patches of harmless flat road surface.

There exists a vast amount of work on dedicated appearance-based detectors for object instances of specific known classes, such as vehicles or pedestrians. Traditionally, these detectors are based on bounding box representations, see for example [Sun et al., 2006, Enzweiler and Gavrila, 2009, Enzweiler et al., 2012, Sivaraman and Trivedi, 2013, Redmon et al., 2015, Liu et al., 2016] and others. If the box-shape assumption approximately holds, impressive performance even at long ranges can be obtained. In [Cordts et al., 2014] the generic Stixel representation is combined with such box-based dedicated object detectors, significantly increasing the detection performance for known object classes compared to the traditional Stixel algorithm.

Moving beyond box-based object detection, recent advances in the pixel-wise semantic classification of full images allow for the use of much richer semantic information in object detection algorithms. In [Scharwächter and Franke, 2015], the pixel-wise classification results are incorporated into the established Stixel framework, effectively combining stereo-based geometry data with color and texture cues exploited via Randomized Decision Forests. Schneider et al. [2016] extend this idea and apply state-of-the-art Fully Convolutional Networks (FCNs) [Long et al., 2015, Cordts et al., 2016] to infer the pixel-wise semantic information used to enhance the Stixel computation.

However, by definition, dedicated classifiers and object detectors are restricted to a limited set of known object classes and are therefore not yet suitable for the task of detecting truly generic and previously unseen object types. Nevertheless, Chapter 5 includes an outlook on how future generic object detection systems may successfully combine geometric modeling approaches with appearance information exploited via modern machine learning techniques.

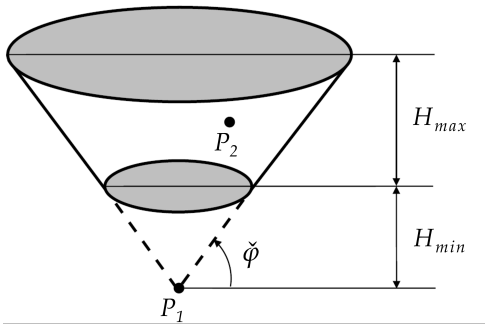


Figure 3.1: Point Compatibility (PC) [Manduchi et al., 2005, Broggi et al., 2011]: Any point P_2 lying within the given truncated cone based at point P_1 is labeled as obstacle and the points are said to be compatible, i.e. are part of an obstacle cluster.

3.1.2 *Point Compatibility*

The so-called Point Compatibility (PC) approach for generic obstacle detection was originally proposed by Manduchi et al. [2005] for autonomous robot navigation, and was later applied by Broggi et al. [2011] in autonomous driving experiments with great success. This geometric obstacle detection method is based on the relative positions of pairs of points in 3D space. Placing a truncated cone on a point P_1 as shown in Fig. 3.1, any point P_2 lying within that cone is labeled as obstacle and said to be compatible with P_1. The cone is defined by the maximum slope angle $\breve{\varphi}$, the minimum relevant obstacle height H_{min} and the maximum connection height threshold H_{max}.

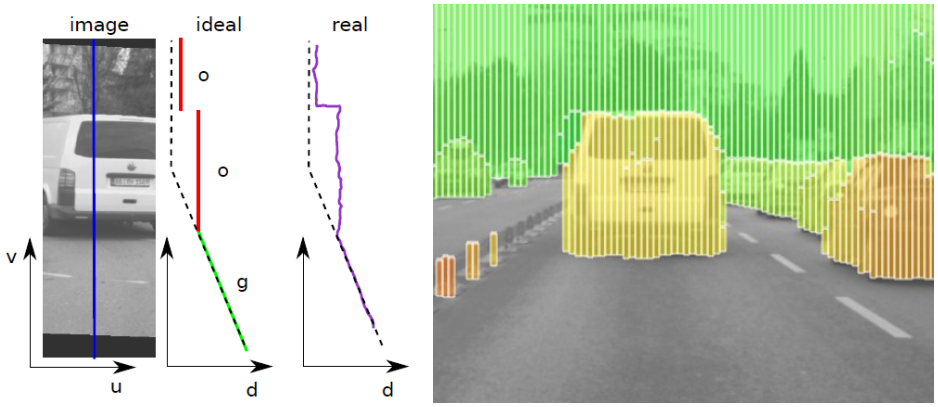
Using a precomputed stereo disparity map as input, all points are tested in this way by traversing the pixels from bottom left to top right. The truncated cones are projected back onto the image plane and the points within the resulting trapezium are labeled accordingly. In this way the algorithm not only provides a pixel-wise obstacle labeling but at the same time performs a meaningful clustering of compatible object points [Broggi et al., 2011].

The PC approach does not depend on any global surface or road model due to its relative geometric decision criterion. However, it does depend directly on the quality of the underlying point cloud.

Due to its flexibility, its convincing performance in previous practical experiments and its generic point-based obstacle representation, the PC approach serves as the first baseline when evaluating the methods presented in this work.

3.1.3 *The Stixel World*

The Stixel algorithm of Badino et al. [2009] and Pfeiffer and Franke [2011] is designed to provide a compact, robust and yet flexible description of 3D scenes, especially in man-made environments with predominantly



(a) Stixel model (ground g, objects o). Image coordinates are denoted by (u,v), with d representing disparity

(b) Stixel output, color-coded by distance

Figure 3.2: Illustration of the column-wise geometric Stixel model formulation of Pfeiffer and Franke [2011], corresponding to a set of 1D segmentation problems in disparity space (a) and example output of non-ground Stixels (b).

horizontal and vertical structures. The algorithm distinguishes between an estimated global ground surface model, in the present case a B-spline model as in [Wedel et al., 2009a], and a set of vertical object segments of variable height (see Fig. 3.2). The corresponding column-wise segmentation task is based directly on a precomputed stereo disparity map and is solved optimally via dynamic programming. By incorporating additional cues such as ordering and gravitational constraints and solving each segmentation task in a globally optimal way, a robust result is obtained. In practice, several image columns are combined in the optimization procedure, yielding Stixels of a certain fixed pixel width (see Fig. 3.2b). As any object can be represented by a variable number of Stixels, the representation can accommodate a large variety of differently shaped and sized objects with adequate accuracy. However, due to the formulation of the algorithm, the quality of the Stixel output directly depends both on the quality of the disparity map as well as the estimated road model.

Since it was first proposed, the Stixel representation has been utilized successfully in a wide range of practical applications within the field of intelligent vehicles and autonomous driving, see for example [Franke et al., 2013]. Therefore, it represents the second main baseline in our evaluation.

3.2 DETECTION BY DIRECT PLANAR HYPOTHESIS TESTING

Considering previous approaches for generic object detection as described in the preceding section, the following critical shortcomings can be identified: Existing algorithms either depend on the validity of simplifying assumptions, such as basic global road surface models, the accuracy of necessary precomputation steps, like dense stereo disparity computation and 3D point cloud reconstruction, or some form of prior knowledge on the visual appearance of expected object classes. Also, due to competing requirements such as robustness and efficiency, the sensitivity of current detection systems is bound to suffer in practice.

Consequently, in the following we present a detection approach which is based on three main design criteria:

1. *Sensitivity*
   In order to be able to reliably detect objects even in challenging cases, e.g. at very long range or particularly small objects at medium ranges, the algorithm has to be sufficiently sensitive. However, at the same time it has to remain robust to real-world conditions, keeping false positive detections at a minimum.

   Precomputation steps such as dense stereo matching algorithms often introduce inaccuracies and reduce sensitivity, for example by neglecting local data-driven evidence in favor of global smoothness constraints. Therefore, the desired approach should minimize intermediate processing steps and extract as much information as possible directly from the input data, i.e. directly from measurable image quantities.

2. *Flexibility*
   Many simplifying model assumptions, such as globally valid basic road models, hold only for a limited set of real-world scenarios. Also, limiting detection capabilities to a set of known object classes is unsuitable for the desired generic detection system. Thus, the detection criteria should be defined to be as flexible as possible and restricting assumptions should be avoided.

3. *Efficiency*
   To allow for the actual application in vehicles with real-time constraints, the computational complexity of any detection system must be kept within the respective bounds. Where possible, the beneficial properties of modern multi-core CPUs and GPUs should be taken into account, favoring easily parallelizable algorithms.

Taking these criteria into account, we propose a novel approach for generic object and obstacle detection called *Direct Planar Hypothesis Testing (PHT)*. We formulate the detection task as a *local* statistical hypothesis testing problem, yielding pixel-wise results based on a

simple but flexible geometric criterion. The approach is designed to utilize the image data of either stereo or multi-view camera setups and operates on a single-frame basis, i.e. without temporal dependencies. In order to avoid a loss in sensitivity by intermediate processing steps, we define generalized likelihood ratio tests with a test statistic based *directly* on a statistical model of the input image data. The null hypothesis $\mathcal{H}_f$ represents observed free-space, while objects correspond to the alternative hypothesis $\mathcal{H}_o$. The tests are performed independently on small local patches distributed across the input images, allowing for straightforward parallelization. Test results are assigned to the pixel at the patch center only if a certain confidence is obtained, hence the output of the algorithm is a semi-dense labeling of pixels which have been classified as obstacles.

In the following, the PHT method is derived and described in detail. Subsequently, Sect. 3.3 demonstrates how the specific properties of standard stereo camera configurations can be exploited to optimize the algorithm for efficiency, yielding a significant speed-up while keeping detection performance at the highest level. Finally, in Sect. 3.4 a post-processing stage is introduced which builds upon the point-wise detection results to generate compact object representations suitable for use in further processing steps of practical applications.

3.2.1 *Geometric Model*

The hypotheses competing in the statistical test are characterized by constraints on the *orientations* of local 3D plane models. In contrast to common global ground or obstacle models, these geometric constraints are designed in a flexible way, allowing for the correct handling of uneven and irregular road profiles. The parameters of each individual local plane are free to vary within certain ranges around a hypothesis reference model. Accordingly, the parameter spaces of $\mathcal{H}_f$ and $\mathcal{H}_o$ are constrained by the angles $\breve{\varphi}_f$ and $\breve{\varphi}_o$, which define the maximum allowed deviation of the respective plane normal vectors from their reference orientations. Without loss of generality, we define the world coordinate system to coincide with the camera coordinate system and set the reference values of the plane normals as parallel to the Y and Z axes of the camera coordinate system, respectively. Given that the stereo camera is roughly aligned with the vehicle frame, this setup approximates a simple flat-ground reference model for the free-space hypothesis and a fronto-parallel object reference model for the obstacle hypothesis. See Fig. 3.3 for an exemplary illustration of the hypothesis models at a hypothetical point located at the origin. Note that in order to handle arbitrarily placed and oriented cameras, the model reference values and constraint angles $\breve{\varphi}_f$ and $\breve{\varphi}_o$ have to be adapted accordingly. The precise values can be tuned based on the expected shape of traversable surfaces and obstacles, respectively.

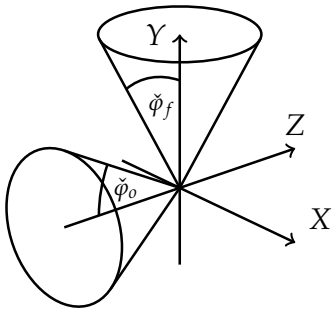


Figure 3.3: The cones defined by $\breve{\varphi}_f$ and $\breve{\varphi}_o$ constrain the permitted plane normal orientations of the free-space and obstacle hypothesis models. The Z axis represents the optical axis of the left camera.

3.2.1.1 *Plane Parameterization and Bound Constraints*

A plane in 3D space has three degrees of freedom and can thus be described by three parameters. To avoid singularities inherent in three-parameter variants [Triggs et al., 2000], each plane is defined by a parameter vector $\vec{\theta} = (n_X, n_Y, n_Z, D)^T$, such that any point $\vec{\tilde{X}}$ lying on the plane satisfies $\vec{\theta}^T \vec{\tilde{X}} = \vec{0}$.

Normalizing the normal vector $\vec{n} = (n_X, n_Y, n_Z)^T$ by enforcing $\|\vec{n}\| = 1$ yields a stable and intuitive representation where $\vec{n}$ represents the orientation of the plane normal and D the normal distance to the origin of the coordinate system. To prevent further ambiguities regarding normal vector orientations, we require that the normal of the free-space hypothesis model always points upward and the normal of the obstacle hypothesis model towards the camera.

By defining the parameterization of the hypothesis models in this way, the constraints on the plane normal angles $\breve{\varphi}_f$ and $\breve{\varphi}_o$ can be directly converted into equivalent simple bound constraints on the respective normal vector components n_X, n_Y and n_Z as follows:

$$-\sin(\breve{\varphi}_f) \leq n_X \leq \sin(\breve{\varphi}_f) \tag{3.1}$$
$$\cos(\breve{\varphi}_f) \leq n_Y \leq 1 \tag{3.2}$$
$$-\sin(\breve{\varphi}_f) \leq n_Z \leq \sin(\breve{\varphi}_f) \tag{3.3}$$

$$-\sin(\breve{\varphi}_o) \leq n_X \leq \sin(\breve{\varphi}_o) \tag{3.4}$$
$$-\sin(\breve{\varphi}_o) \leq n_Y \leq \sin(\breve{\varphi}_o) \tag{3.5}$$
$$-1 \leq n_Z \leq -\cos(\breve{\varphi}_o) \tag{3.6}$$

We further add weak bounds on the normal distance parameter D to avoid singularities and ensure reasonable results:

$$0 < D < D_{max}. \tag{3.7}$$

3.2.2 *Hypothesis Test*

As described in Sect. 2.2.1, the design of a likelihood ratio test yielding a provably optimal detector would require full knowledge of the PDFs and all parameters of the data model under $\mathcal{H}_f$ and $\mathcal{H}_o$. As this information is not available, we make use of the GLRT as defined in (2.26) instead. For each hypothesis $\mathcal{H}_i = \mathcal{H}_{\{f,o\}}$ the unknown parameters $\vec{\theta}_i$ are replaced by their MLEs $\hat{\vec{\theta}}_i$, assuming the respective hypothesis to be true. We then decide for the alternative hypothesis $\mathcal{H}_o$ only if

$$L_G(\vec{\mathcal{I}}) = \frac{p(\vec{\mathcal{I}}; \hat{\vec{\theta}}_o, \mathcal{H}_o)}{p(\vec{\mathcal{I}}; \hat{\vec{\theta}}_f, \mathcal{H}_f)} > \gamma, \tag{3.8}$$

or equivalently if

$$\ln\left(p(\vec{\mathcal{I}}; \hat{\vec{\theta}}_o, \mathcal{H}_o)\right) - \ln\left(p(\vec{\mathcal{I}}; \hat{\vec{\theta}}_f, \mathcal{H}_f)\right) > \ln(\gamma), \tag{3.9}$$

where the generalized likelihood ratio $L_G(\vec{\mathcal{I}})$ represents the test statistic and $\vec{\mathcal{I}}$ is the intensity data vector of the stereo image pair.

Note that for patches classified as obstacle, the MLE $\hat{\vec{\theta}}_o$ implicitly provides an optimized estimate of the obstacle position in 3D space.

Since the formulated hypothesis models do not allow to employ a simple parameter test as described in Sect. 2.2.2.1, the requirements to analytically determine the optimal decision threshold are not fulfilled. Therefore, we determine the optimal value of γ from an empirical analysis of the detection performance on relevant data, see Sect. 3.5.4.

3.2.3 *Data Model*

We formulate a statistical image formation model to define the likelihood terms in (3.9). The discretized intensity values $I_l(\vec{x}_l)$ and $I_r(\vec{x}_r)$ within the patch area Ω in the left and right image are considered as noisy samples of the observed continuous image intensity signal f at position $\vec{x}$, where $\vec{x} = \vec{x}_l$. The terms $\alpha_l(\vec{x})$ and $\alpha_r(\vec{x})$ model a potential local intensity offset, while $\eta_l(\vec{x})$ and $\eta_r(\vec{x})$ represent samples from a noise distribution with zero mean and an assumed variance σ^2. As the left camera of the stereo setup is specified as the reference camera defining the origin of the coordinate system, we get:

$$I_l(\vec{x}) = f(\vec{x}) + \alpha_l(\vec{x}) + \eta_l(\vec{x}) \tag{3.10}$$

$$I_r\left(\vec{W}(\vec{x}, \vec{\theta})\right) = f(\vec{x}) + \alpha_r(\vec{x}) + \eta_r(\vec{x}). \tag{3.11}$$

The warp $\vec{W}(\vec{x}, \vec{\theta}) = \left(W_x(\vec{x}, \vec{\theta}), W_y(\vec{x}, \vec{\theta})\right)^T$ transforms the image coordinates $\vec{x}$ from the left to the right image via the camera parameters

$\mathbf{P}_r = \mathbf{K}_r \left(\mathbf{R}_r \,|\, \vec{t}_r \right)$. It is parameterized by the vector $\vec{\theta}$, which represents the geometric model of the true hypothesis. For the used locally planar models the warp corresponds to a multiplication by the homography [Hartley and Zisserman, 2004]

$$\mathbf{H} = \mathbf{K}_r \left(\mathbf{R}_r - \frac{1}{D} \vec{t}_r \, \vec{n}^T \right) \mathbf{K}_l^{-1}. \tag{3.12}$$

First, to compensate for a potential local offset α, the mean intensity of the considered patch is removed from all pixels in the patch area Ω in each image. Then, treating the intensity values of all pixels in Ω as observations of f with additive i.i.d. noise, we get from (3.11):

$$\ln\left(p(\vec{\mathcal{I}}; \vec{\theta}, \mathcal{H})\right) = \sum_{\vec{x} \in \Omega} C_1 - C_2 \cdot \rho\left(I_r\left(\vec{W}(\vec{x}, \vec{\theta})\right) - f(\vec{x})\right), \tag{3.13}$$

where C_1 and C_2 are constants and ρ represents the characteristic loss function of the assumed noise model, e.g. a quadratic loss for a normal distribution or a L1 loss for a Laplacian distribution.

The relevant contribution of each hypothesis to (3.9) then reduces to a sum of pixel-wise residuals r over the local patch area Ω, which we call the cost function F:

$$F_i(\vec{\theta}_i) = \sum_{\vec{x} \in \Omega} \rho\left(r(\vec{x}, \vec{\theta}_i)\right) = \sum_{\vec{x} \in \Omega} \rho\left(I_r\left(\vec{W}(\vec{x}, \vec{\theta}_i)\right) - f(\vec{x})\right). \tag{3.14}$$

Finding the MLE of the parameter values then corresponds to minimizing the negative log-likelihood of each hypothesis, i.e. the respective cost function. This task is represented by the non-linear optimization problem with simple bound constraints

$$\begin{aligned}
\hat{\vec{\theta}}_i &= \arg\min_{\vec{\theta}_i} \left(-\ln\left(p(\vec{\mathcal{I}}; \vec{\theta}_i, \mathcal{H}_i)\right)\right) \\
&= \arg\min_{\vec{\theta}_i} \sum_{\vec{x} \in \Omega} \rho\left(I_r\left(\vec{W}(\vec{x}, \vec{\theta}_i)\right) - f(\vec{x})\right) \\
&= \arg\min_{\vec{\theta}_i} \sum_{\vec{x} \in \Omega} \rho\left(r(\vec{x}, \vec{\theta}_i)\right) \\
&= \arg\min_{\vec{\theta}_i} \left(F_i(\vec{\theta}_i)\right) \\
&\text{s.t.} \quad |\varphi_i| \leq \breve{\varphi}_i.
\end{aligned} \tag{3.15}$$

The angular bound constraints are substituted by the corresponding bounds on the actual parameter vector as given in (3.1)-(3.7).

3.2.4 *Optimization*

The optimization procedure is performed using the constrained Levenberg-Marquardt algorithm described in Sect. 2.1.4 (Alg. 2.1). Since the procedure is identical for free-space and obstacle hypotheses except for the values used to specify constraints and initial parameters, in the following the class index $i = \{f, o\}$ is omitted for brevity.

As described in Sect. 2.1, in order to reduce the value of the cost function F, in each iteration a parameter update $\Delta\vec{\theta}$ is applied to $\vec{\theta}$ using the update step operator

$$\vec{\theta} \leftarrow \vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}). \tag{3.16}$$

The update vector is computed by solving

$$-\vec{g}^T(\vec{\theta}) = \mathbf{A}_{LM}(\vec{\theta})\Delta\vec{\theta}, \tag{3.17}$$

where $\vec{g}$ is the gradient of the cost function with respect to the parameter update and $\mathbf{A}_{LM}$ is the approximate Levenberg-Marquardt Hessian as defined in (2.16). At each iteration the optimization procedure therefore requires the evaluation of the gradient as well as the approximate Hessian of the cost function defined in (3.14).

3.2.4.1 *Local Parameterization*

While Sect. 3.2.1.1 motivates the choice of a stable global parameterization of the hypothesis plane models, the inherent overparameterization adds a redundant dimension to the domain of the cost function and hence to the search space of the optimization problem. Therefore, we introduce a minimal local parameterization to obtain a more efficient and stable optimization procedure [Triggs et al., 2000]. While singularities might occur with this minimal parameterization when applied globally, it provides very robust and accurate results locally, as shown by Baker et al. [2006]. For each step, an update vector $\Delta\vec{\theta}^L$ of lower dimension than $\vec{\theta}$ is used, corresponding to the unnormalized normal vector update

$$\Delta\vec{\theta}^L = \Delta\vec{n} = (\Delta n_X, \Delta n_Y, \Delta n_Z)^T. \tag{3.18}$$

Since the global plane parameterization enforces $\|\vec{n}\| = 1$, the valid parameter space of $\vec{n}$ is restricted to the unit sphere. During optimization, all parameter steps are automatically projected back onto the unit sphere surface via $\vec{\oplus}$. In this way, an indirect update of the remaining parameter D is obtained:

$$\vec{n} \leftarrow \frac{\vec{n} + \Delta\vec{n}}{\|\vec{n} + \Delta\vec{n}\|}, \qquad D \leftarrow \frac{D}{\|\vec{n} + \Delta\vec{n}\|} \tag{3.19}$$

Note that the update operator fulfills the identity relationship $\vec{\oplus}(\vec{\theta}, \vec{0}) = \vec{\theta}$.

3.2.4.2 *Gradient*

Analogous to (2.9), the gradient and hence the Jacobian $\mathbf{J}_{F\circ\vec{\oplus}}$ of the cost function with respect to the local parameter update $\Delta\vec{\theta}^{\,L}$ is

$$\vec{g}^{\,T}\left(\vec{\oplus}(\vec{\theta},\Delta\vec{\theta}^{\,L})\right) = \mathbf{J}_{F\circ\vec{\oplus}}(\vec{\theta},\Delta\vec{\theta}^{\,L}) = \mathbf{J}_F(\vec{\oplus}(\vec{\theta},\Delta\vec{\theta}^{\,L}))\,\mathbf{J}_{\vec{\oplus}}(\Delta\vec{\theta}^{\,L}), \tag{3.20}$$

where $\mathbf{J}_F$ denotes the Jacobian of the cost function with respect to the global parameter vector and $\mathbf{J}_{\vec{\oplus}}$ represents the Jacobian of the parameter update step with respect to the local parameterization. Here, $\mathbf{J}_F$ can be expressed analytically as follows:

$$\begin{aligned}
&\mathbf{J}_F(\vec{\oplus}(\vec{\theta},\Delta\vec{\theta}^{\,L})) \\
&\quad= \frac{\partial F(\vec{\oplus}(\vec{\theta},\Delta\vec{\theta}^{\,L}))}{\partial(\oplus_{n_X}(\vec{\theta},\Delta\vec{\theta}^{\,L}),\oplus_{n_Y}(\vec{\theta},\Delta\vec{\theta}^{\,L}),\oplus_{n_Z}(\vec{\theta},\Delta\vec{\theta}^{\,L}),\oplus_D(\vec{\theta},\Delta\vec{\theta}^{\,L}))} \\
&\quad= \frac{\partial F(\vec{\oplus}(.))}{\partial(\oplus_{n_X}(.),\ldots,\oplus_D(.))} \\
&\quad= \frac{\partial \sum\limits_{\vec{x}\in\Omega}\rho\left(r(\vec{x},\vec{\oplus}(.))\right)}{\partial(\oplus_{n_X}(.),\ldots,\oplus_D(.))} \\
&\quad= \sum_{\vec{x}\in\Omega}\rho'\left(r(\vec{x},\vec{\oplus}(.))\right)\frac{\partial r(\vec{x},\vec{\oplus}(.))}{\partial(\oplus_{n_X}(.),\ldots,\oplus_D(.))},
\end{aligned} \tag{3.21}$$

where ρ' denotes the derivative of the loss function, computed as described in Sect. 2.1.5. The partial derivatives of the residuals with respect to the components of $\vec{\oplus}$ for each pixel within the patch are

$$\begin{aligned}
\frac{\partial r(\vec{x},\vec{\oplus}(.))}{\partial\left(\oplus_{n_X}(.),\ldots,\oplus_D(.)\right)} &= \frac{\partial\left(I_r\left(\vec{W}(\vec{x},\vec{\oplus}(.))\right)-f(\vec{x})\right)}{\partial\left(\oplus_{n_X}(.),\ldots,\oplus_D(.)\right)} \\
&= \frac{\partial I_r\left(\vec{W}(\vec{x},\vec{\oplus}(.))\right)}{\partial\left(W_x(\vec{x},\vec{\oplus}(.)),W_y(\vec{x},\vec{\oplus}(.))\right)}\frac{\partial\left(W_x(\vec{x},\vec{\oplus}(.)),W_y(\vec{x},\vec{\oplus}(.))\right)}{\partial(\oplus_{n_X}(.),\ldots,\oplus_D(.))}.
\end{aligned} \tag{3.22}$$

Here the first term represents the image gradient of I_r, i.e. the partial derivatives of the image intensities in horizontal and vertical direction, evaluated at $\vec{W}(\vec{x},\vec{\oplus}(\vec{\theta},\Delta\vec{\theta}^{\,L}))$:

$$\begin{aligned}
\frac{\partial I_r\left(\vec{W}(\vec{x},\vec{\oplus}(.))\right)}{\partial\left(W_x(\vec{x},\vec{\oplus}(.)),W_y(\vec{x},\vec{\oplus}(.))\right)} &= \begin{pmatrix}\frac{\partial I_r\left(\vec{W}(\vec{x},\vec{\oplus}(.))\right)}{\partial W_x(\vec{x},\vec{\oplus}(.))} \\ \frac{\partial I_r\left(\vec{W}(\vec{x},\vec{\oplus}(.))\right)}{\partial W_y(\vec{x},\vec{\oplus}(.))}\end{pmatrix}^T \\
&= \vec{\nabla} I_r^T\left(\vec{W}(\vec{x},\vec{\oplus}(\vec{\theta},\Delta\vec{\theta}^{\,L}))\right).
\end{aligned} \tag{3.23}$$

The Jacobian of the image warp $\vec{W}$ itself is

$$\begin{aligned}&\frac{\partial\left(W_x(\vec{x},\vec{\oplus}(.)),W_y(\vec{x},\vec{\oplus}(.))\right)}{\partial(\oplus_{n_X}(.),\ldots,\oplus_D(.))}\\&=\frac{1}{(\mathbf{H}_{3,:}\vec{x})^2}\begin{pmatrix}\frac{\partial(\mathbf{H}_{1,:}\vec{x})}{\partial(\oplus_{n_X}(.),\ldots,\oplus_D(.))}\mathbf{H}_{3,:}\vec{x}-\frac{\partial(\mathbf{H}_{3,:}\vec{x})}{\partial(\oplus_{n_X}(.),\ldots,\oplus_D(.))}\mathbf{H}_{1,:}\vec{x}\\\frac{\partial(\mathbf{H}_{2,:}\vec{x})}{\partial(\oplus_{n_X}(.),\ldots,\oplus_D(.))}\mathbf{H}_{3,:}\vec{x}-\frac{\partial(\mathbf{H}_{3,:}\vec{x})}{\partial(\oplus_{n_X}(.),\ldots,\oplus_D(.))}\mathbf{H}_{2,:}\vec{x}\end{pmatrix},\end{aligned}\tag{3.24}$$

where $\mathbf{H}_{i,:}$ represents the ith row of the plane-induced homography $\mathbf{H}$. The partial derivatives of the homography matrix are in turn[1]

$$\frac{\partial\mathbf{H}}{\partial n_X}=\mathbf{K}_r\left(-\frac{1}{D}\vec{t}_r\,(1\;0\;0)\right)\mathbf{K}_l^{-1}\tag{3.25}$$

$$\frac{\partial\mathbf{H}}{\partial n_Y}=\mathbf{K}_r\left(-\frac{1}{D}\vec{t}_r\,(0\;1\;0)\right)\mathbf{K}_l^{-1}\tag{3.26}$$

$$\frac{\partial\mathbf{H}}{\partial n_Y}=\mathbf{K}_r\left(-\frac{1}{D}\vec{t}_r\,(0\;0\;1)\right)\mathbf{K}_l^{-1}\tag{3.27}$$

$$\frac{\partial\mathbf{H}}{\partial D}=\mathbf{K}_r\left(\frac{1}{D^2}\vec{t}_r\,\vec{n}^T\right)\mathbf{K}_l^{-1}.\tag{3.28}$$

The Jacobian $\mathbf{J}_{\vec{\oplus}}$ of the parameter update step with respect to $\Delta\vec{\theta}^L$ takes the form

$$\mathbf{J}_{\vec{\oplus}}(\Delta\vec{\theta}^L)=\frac{\partial(\oplus_{n_X}(.),\ldots,\oplus_D(.))}{\partial(\Delta n_X,\Delta n_Y,\Delta n_Z)}\tag{3.29}$$

$$=\begin{pmatrix}\frac{\partial\oplus_{n_X}(\vec{\theta},\Delta\vec{\theta}^L)}{\partial\Delta n_X}&\frac{\partial\oplus_{n_X}(\vec{\theta},\Delta\vec{\theta}^L)}{\partial\Delta n_Y}&\frac{\partial\oplus_{n_X}(\vec{\theta},\Delta\vec{\theta}^L)}{\partial\Delta n_Z}\\\frac{\partial\oplus_{n_Y}(\vec{\theta},\Delta\vec{\theta}^L)}{\partial\Delta n_X}&\frac{\partial\oplus_{n_Y}(\vec{\theta},\Delta\vec{\theta}^L)}{\partial\Delta n_Y}&\frac{\partial\oplus_{n_Y}(\vec{\theta},\Delta\vec{\theta}^L)}{\partial\Delta n_Z}\\\frac{\partial\oplus_{n_Z}(\vec{\theta},\Delta\vec{\theta}^L)}{\partial\Delta n_X}&\frac{\partial\oplus_{n_Z}(\vec{\theta},\Delta\vec{\theta}^L)}{\partial\Delta n_Y}&\frac{\partial\oplus_{n_Z}(\vec{\theta},\Delta\vec{\theta}^L)}{\partial\Delta n_Z}\\\frac{\partial\oplus_D(\vec{\theta},\Delta\vec{\theta}^L)}{\partial\Delta n_X}&\frac{\partial\oplus_D(\vec{\theta},\Delta\vec{\theta}^L)}{\partial\Delta n_Y}&\frac{\partial\oplus_D(\vec{\theta},\Delta\vec{\theta}^L)}{\partial\Delta n_Z}\end{pmatrix},\tag{3.30}$$

where

$$\mathbf{J}_{\vec{\oplus}}(\Delta\vec{\theta}^L)\Big|_{\Delta\vec{\theta}^L=\vec{0}}=\begin{pmatrix}1-n_X^2&-n_Xn_Y&-n_Xn_Z\\-n_Yn_X&1-n_Y^2&-n_Yn_Z\\-n_Zn_X&-n_Zn_Y&1-n_Z^2\\-Dn_X&-Dn_Y&-Dn_Z\end{pmatrix}.\tag{3.31}$$

Using these analytic results, the gradient of the cost function can readily be evaluated.

1 Similar results have been reported by Kähler and Denzler [2012] for the related problem of direct piecewise-planar structure-from-motion estimation.

3.2.4.3 *Approximate Hessian*

The Hessian is obtained by computing the partial derivatives of the cost function gradient with respect to the components of $\vec{\theta}^L$. Following Sect. 2.1.2 we combine (3.21) and (3.29) to create the auxiliary variable

$$\mathbf{J}^*(\vec{x}, \vec{\theta}, \Delta\vec{\theta}^L) = \frac{\partial r(\vec{x}, \vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}^L))}{\partial(\oplus_{n_X}(.), \ldots, \oplus_D(.))} \frac{\partial(\oplus_{n_X}(.), \ldots, \oplus_D(.))}{\partial(\Delta n_X, \Delta n_Y, \Delta n_Z)}. \tag{3.32}$$

The Hessian can then be written as

$$\begin{aligned} &\mathbf{A}(\vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}^L)) \\ &\quad = \sum_{\vec{x}\in\Omega} \rho'' \left(r(\vec{x}, \vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}^L)) \right) \cdot \mathbf{J}^{*\,T}(\vec{x}, \vec{\theta}, \Delta\vec{\theta}^L) \mathbf{J}^*(\vec{x}, \vec{\theta}, \Delta\vec{\theta}^L) \\ &\quad + \sum_{\vec{x}\in\Omega} \rho' \left(r(\vec{x}, \vec{\oplus}(\vec{\theta}, \Delta\vec{\theta}^L)) \right) \cdot \frac{\partial(\mathbf{J}^*_{\Delta n_X}(.), \ldots, \mathbf{J}^*_{\Delta n_Z}(.))}{\partial(\Delta n_X, \Delta n_Y, \Delta n_Z)}. \end{aligned} \tag{3.33}$$

Dropping all second-order derivatives and evaluating at $\Delta\vec{\theta}^L = \vec{0}$, we obtain the approximate Gauss-Newton Hessian

$$\mathbf{A}_{GN}(\vec{\theta}) = \sum_{\vec{x}\in\Omega} \rho'' \left(r(\vec{x}, \vec{\theta}) \right) \cdot \mathbf{J}^{*\,T}(\vec{x}, \vec{\theta}, \vec{0}) \mathbf{J}^*(\vec{x}, \vec{\theta}, \vec{0}) \tag{3.34}$$

and consequently the approximate Levenberg-Marquardt Hessian

$$\mathbf{A}_{LM}(\vec{\theta}) = \mathbf{A}_{GN}(\vec{\theta}) + \gamma \mathsf{diag} \left\{ \mathbf{A}_{GN}(\vec{\theta}) \right\}. \tag{3.35}$$

3.2.4.4 *Symmetric Residuals*

Considering the image formation model defined in Sect. 3.2.3 and the resulting formulation of the optimization procedure, one problem still remains open: the residuals depend on the unknown image signal f. Commonly, in direct differential matching approaches based on the seminal work of Lucas and Kanade [1981], this problem is circumvented by a slight reformulation of the data model: Equation (3.10) is solved for the unknown image signal f and the result is plugged into (3.11). This yields the correspondence relation

$$I_r \left(\vec{W}(\vec{x}, \vec{\theta}) \right) = I_l(\vec{x}) + \alpha(\vec{x}) + \eta(\vec{x}) \tag{3.36}$$

and resulting residual terms of the form

$$r(\vec{x}, \vec{\theta}) = I_r \left(\vec{W}(\vec{x}, \vec{\theta}) \right) - I_l(\vec{x}). \tag{3.37}$$

As an alternative, inspired by Mester [2014] we propose a pragmatic approach to simultaneously estimate the unknown image signal f in conjunction with the optimal warp parameters. Given that the warp parameters are sufficiently close to the correct solution, in the simplest form

this is done by taking the pixel-wise mean of the concurrently realigned input image patches in each iteration:

$$\hat{f}(\vec{x}) = \frac{I_l(\vec{x}) + I_r\left(\vec{W}(\vec{x},\vec{\theta})\right)}{2}. \tag{3.38}$$

Plugging the estimate $\hat{f}$ into (3.11) we arrive at the desired residual terms, now independent of the unknown f. Note that this approach gains relevance when considering multi-view applications, since using (3.36) results in asymmetric residuals with a bias towards the reference image. Here the benefit of using the estimate $\hat{f}$ becomes apparent as the bias towards the reference image is mitigated. See Sect. 3.2.6 for further details.

3.2.4.5 *Conditioning*

The described optimization procedure can only provide suitable results if the solution to (3.17) yields accurate update vectors which are descent directions for (3.14). In order to ensure the required positive definite approximate Hessian and a well-conditioned system, we specify a lower threshold on the minimum eigenvalue of $\mathbf{A}_{GN}$. In practice, this means that the optimization is only carried out for patches in sufficiently textured image areas and thus only reliable decisions are reported as output of the likelihood ratio test. A similar filtering approach was already applied in the seminal work of Tomasi and Kanade [1991].

3.2.4.6 *Initialization*

Since finding the global optimum of the parameter values cannot be guaranteed in general, a suitable initialization of $\vec{\theta}$ is necessary. We initialize the free-space models from a coarse global ground plane estimate and obstacles from fronto-parallel plane models at a certain initial distance. The required initial distance values are extracted from a coarse, pre-computed disparity map. The efficacy of this simple initialization scheme is confirmed in the practical experiments of Sect. 3.5.

3.2.5 *Model Consistency*

After a decision has been obtained from the hypothesis test, the winning hypothesis is rechecked for consistency with the assumed underlying image formation model of Sect. 3.2.3.

First, we require that the number of outlier residuals generated by the winning hypothesis model does not exceed a given threshold, e. g. 50% of the pixels of the patch area Ω. Outliers are defined by a residual magnitude of $|r| > 3\sigma$, where σ^2 is the variance of the assumed image noise distribution.

Second, the residual sample mean $\bar{r}$ is considered. Here we expect $\bar{r} = 0$ with variance $\frac{\sigma^2}{|\Omega_i|}$, with $|\Omega_i|$ denoting the number of pixels classified as

inliers in the first step. If the absolute value of the inlier residual sample mean is larger than $3\sqrt{\frac{\sigma^2}{|\Omega_i|}}$, the patch is discarded.

Finally, we analyze the sample variance of the patch residuals, which is computed as $\sigma_r^2 = \frac{1}{|\Omega_i|-1} \sum_{\vec{x}\in\Omega_i} (r(\vec{x}) - \bar{r})^2$. Here we require $\sigma_r < 3\sigma$ for the result to be accepted.

Of course, the acceptance criteria used in the consistency checks can be adapted to the application scenario and respective requirements. For example, the threshold values might be relaxed in order to maximize detection rate and retain even more point detections at object borders. However, this may come at the cost of an increased number of false positives just outside object borders, resulting in so-called foreground fattening artifacts.

Fig. 3.4 illustrates the output of the presented PHT algorithm in two exemplary scenes. In Fig. 3.4b, the result of the hypothesis test is shown as a color-coded overlay of the center pixel of each considered patch. Patches retaining the free-space hypothesis are shown in green, whereas obstacle decisions are depicted in red. Note that retaining the free-space hypothesis does not guarantee the existence of free-space at a given point, it simply indicates that the evidence provided by the data in favor of the alternative obstacle hypothesis is not sufficient to reject the null hypothesis. Patches which result in an ill-conditioned system or fail the model consistency checks are shown in gray.

In Fig. 3.4c, the optimized distance estimates of the detected obstacle points are illustrated, where close points are shown in red and distant ones in green.

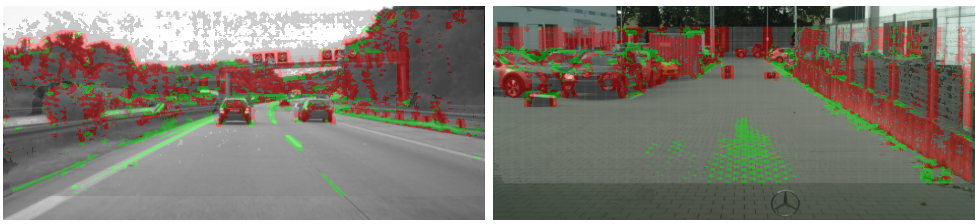
3.2.6 *Generalization to Multi-View Configurations*

Due to the generic geometric formulation of the PHT detection approach, a generalization from the two-view stereo setup described above to calibrated multi-view setups is straightforward and, in fact, a quite natural conceptual step. Several benefits of utilizing suitable multi-view setups for obstacle detection have previously been demonstrated by Williamson and Thorpe [1999].

Considering a camera setup of one reference view and M additional views with known intrinsic and extrinsic parameters, the data of all views can be exploited jointly in both the parameter optimization procedure and the subsequent likelihood ratio test[2].

2 The stereo setup, which is the focus of this work, is a special case with $M = 1$.

(a) Left stereo input image



(b) Hypothesis test result

(c) Color-coded distance estimates of the detected obstacle points

Figure 3.4: Exemplary output of the PHT detection algorithm in a highway (left) and urban (right) setting. As shown in (b), the hypothesis test either retains the free-space hypothesis (green) or rejects it in favor of the obstacle hypothesis (red). No decision is made at locations with unreliable data (gray).

A cost function analogous to (3.14) can be formulated as follows:

$$F(\vec{\theta}) = \sum_{m=1..M} \sum_{\vec{x}\in\Omega} \rho\left(r_m(\vec{x},\vec{\theta})\right) \tag{3.39}$$

$$= \sum_{m=1..M} \sum_{\vec{x}\in\Omega} \rho\left(I_m\left(\vec{W}_m(\vec{x},\vec{\theta})\right) - f(\vec{x})\right). \tag{3.40}$$

The residuals r_m and the corresponding derivatives are computed by directly aggregating over the patch area in each respective input image. Each warp $\vec{W}_m$ and the corresponding derivatives are defined by the 3D plane model parameters $\vec{\theta}$ of the considered hypothesis and the intrinsic and extrinsic parameters $\mathbf{P}_m = \mathbf{K}_m\left[\mathbf{R}_m \,|\, \vec{t}_m\right]$ of view m. The optimal parameter values are then obtained as described in Sect. 3.2.4.

Following Sect. 3.2.4.4, the unknown f can be approximated by

$$\hat{f}(\vec{x}) = \frac{1}{M+1}\left(I_0(\vec{x}) + \sum_{m=1..M} I_m\left(\vec{W}_m(\vec{x},\vec{\theta})\right)\right), \tag{3.41}$$

where I_0 denotes the reference image. Note that the estimate $\hat{f}$ can be expected to become more accurate in the multi-view case as more samples are available to average out noise and suppress outliers.

However, from a practical point of view, especially in large-scale automotive applications the additional requirements of multi-view setups regarding hardware expenses, computational complexity as well as calibration effort might prove to be prohibitive for the time being.

3.3 FAST DIRECT PLANAR HYPOTHESIS TESTING

The PHT method presented in the previous section provides high flexibility in terms of both model parameters and camera configurations, including multi-view setups. However, for calibrated stereo cameras a simplified parametrization can be utilized, reducing the number of free parameters and the complexity of the optimization problem as well as avoiding the need for all intermediate point projection operations.

Therefore we propose Fast Direct Planar Hypothesis Testing (FPHT), a method that exploits such a reparametrization, resulting in a significant computational speed-up without sacrificing detection performance in practice.

3.3.1 *Reparametrization in Disparity Space*

The proposed reparametrization is based on considering only

- rectified stereo image pairs and
- plane models without yaw or roll angles, i.e. $n_X = 0$.

Under these assumptions, computation of the warp $\vec{W}$ can be simplified significantly, since a plane with $n_X = 0$ can be represented by a line in stereo disparity space [Labayrade et al., 2002]. The new parameter vector $\vec{\theta} = (a, b)^T$ consists only of the disparity slope a and the offset b:

$$\vec{W}(\vec{x}, \vec{\theta}) = \begin{pmatrix} x - d \\ y \end{pmatrix} = \begin{pmatrix} x - (a\bar{y} + b) \\ y \end{pmatrix}. \tag{3.42}$$

Disparity is denoted by d, while $\bar{y} = \frac{y_c - y}{h/2}$ represents normalized vertical image coordinates, with the patch center position y_c and patch height h. Disparity slope and offset directly relate to the 3D plane parameters as

$$a = -n_Y \frac{f_x}{f_y} \frac{B}{D} \tag{3.43}$$

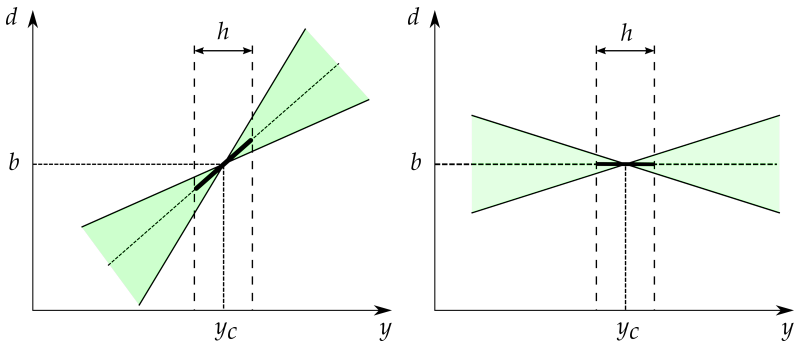
$$b = -\frac{B}{D} \left(n_Y (y_c - y_0) \frac{f_x}{f_y} + n_Z f_x \right), \tag{3.44}$$

where the stereo camera's focal lengths, vertical principal point and baseline length are denoted as f_x, f_y, y_0 and B, respectively.

This minimal parameterization directly represents the remaining two degrees of freedom of the planar model, eliminating the need for a local parameterization as in Sect. 3.2.4.1.

3.3.2 *Bound Constraints*

In contrast to the PHT method, where all 3D plane parameters are bounded by globally valid box constraints (see (3.1)-(3.7)), we have to



(a) Free-space hypothesis (b) Obstacle hypothesis

Figure 3.5: Illustration of the planar model representation and exemplary bound constraints in disparity space for a given image patch of height h centered at (x_c, y_c). Feasible regions are shown in green.

consider the fact that two planes with *the same* slope angle φ_i in 3D space, but with different normal distances D to the camera, will in general have different disparity offsets b *as well as* different slopes a in disparity space. It is therefore not possible to derive independent global bounds on a and b from the original bounds on the 3D plane parameters. Instead, we formulate a joint constraint on the plane orientation by plugging the original bounds on the normal vector $\vec{n}$ into the linear relation

$$a = \frac{b}{(y_0 - y_c) + f_y \frac{n_Z}{n_Y}} = b \cdot const. \tag{3.45}$$

If an optimization step violates the bound and results in an invalid configuration of a and b, both values are warped back onto the bounding line via vector projection as described in Sect. 2.1.4. This constraint is supplemented by upper and lower bounds on the disparity offset b, similar to the normal distance checks in (3.7).

Fig. 3.5 illustrates the planar model representation in disparity space employed by FPHT, including exemplary feasible regions for varying slope a, assuming a constant offset b.

3.3.3 *Inverse Compositional Optimization*

While the proposed simplified parameterization already results in a speed-up by eliminating free parameters and reducing the complexity of warping operations, the formulation of the core optimization procedure itself can be further improved as well.

Considering the optimization procedure as described in Sect. 3.2.4, it becomes apparent that the approximation of the Hessian $\mathbf{A}_{LM}$ directly depends on the current estimate of $\vec{\theta}$ and thus has to be recomputed in every iteration. Furthermore, the image gradients have to be recomputed and warped to sub-pixel positions in every iteration as well. This does

not only reduce the algorithm's computational efficiency, but can also introduce errors due to imperfect gradient filters and interpolation functions [Sutton et al., 2009]. Baker and Matthews [2004] offer a detailed analysis of such efficiency issues and propose a reformulation of the optimization procedure to eliminate these drawbacks, the so-called inverse compositional algorithm.

In the following, the concept of warp composition is introduced by applying the forward compositional formulation to the optimization problem at hand. Subsequently, the inverse compositional formulation of the FPHT algorithm is derived.

3.3.3.1 *Forward Warp Composition*

First, the update operator $\vec{\oplus}$ is replaced by the composition operator $\vec{\odot}$, which is applied to the warp $\vec{W}$ instead of to the parameter vector $\vec{\theta}$. Consequently, in each iteration we now apply incremental warp updates $\vec{W}(\vec{x}, \Delta\vec{\theta})$ to the current warp $\vec{W}(\vec{x}, \vec{\theta})$ via

$$\vec{W}(\vec{x}, \vec{\theta}) \leftarrow \vec{\odot}\left(\vec{W}(\vec{x}, \vec{\theta}), \vec{W}(\vec{x}, \Delta\vec{\theta})\right), \tag{3.46}$$

with

$$\vec{\odot}\left(\vec{W}(\vec{x}, \vec{\theta}), \vec{W}(\vec{x}, \Delta\vec{\theta})\right) \equiv \vec{W}\left(\vec{W}(\vec{x}, \Delta\vec{\theta}), \vec{\theta}\right) \tag{3.47}$$

$$= \begin{pmatrix} x - (\Delta a\bar{y} + \Delta b) - (a\bar{y} + b) \\ y \end{pmatrix} \tag{3.48}$$

and the identity warp $\vec{W}(\vec{x}, \vec{0}) = \vec{x}$. For a more concise notation, in the following the current warp $\vec{W}(\vec{x}, \vec{\theta})$ will be simply denoted as $\vec{W}$:

$$\vec{W}(\vec{x}, \vec{\theta}) \equiv \vec{W}. \tag{3.49}$$

Rewriting the cost function of (3.14) as a function of the warp $\vec{W}$ yields

$$F(\vec{W}) = \sum_{\vec{x}\in\Omega} \rho\left(r(\vec{x}, \vec{W})\right) = \sum_{\vec{x}\in\Omega} \rho\left(I_r(\vec{W}) - f(\vec{x})\right). \tag{3.50}$$

The gradient of the cost function with respect to the parameter update $\Delta\vec{\theta} = (\Delta a, \Delta b)^T$, which is equivalent to the Jacobian $\mathbf{J}_{F\circ\vec{\odot}\circ\vec{W}}$, is then

$$\begin{aligned} \vec{g}^T\left(\vec{\odot}\left(\vec{W}, \vec{W}(\vec{x}, \Delta\vec{\theta})\right)\right) &= \mathbf{J}_{F\circ\vec{\odot}\circ\vec{W}}\left(\vec{\odot}\left(\vec{W}, \vec{W}(\vec{x}, \Delta\vec{\theta})\right)\right) \\ &= \mathbf{J}_F\left(\vec{\odot}\left(\vec{W}, \vec{W}(\vec{x}, \Delta\vec{\theta})\right)\right)\mathbf{J}_{\vec{\odot}}\left(\vec{W}(\vec{x}, \Delta\vec{\theta})\right)\mathbf{J}_{\vec{W}}(\Delta\vec{\theta}), \end{aligned} \tag{3.51}$$

where $\mathbf{J}_F$ is the Jacobian of the cost function with respect to the composed warp, $\mathbf{J}_{\vec{\odot}}$ is the Jacobian of the composition operator with respect to the incremental warp update, and $\mathbf{J}_{\vec{W}}$ is the Jacobian of the image warp with respect to the parameter update $\Delta\vec{\theta}$.

The Jacobian $\mathbf{J}_F$ can be expressed as

$$\mathbf{J}_F\left(\vec{\odot}(.)\right) = \sum_{\vec{x}\in\Omega} \rho'\left(r\left(\vec{x}, \vec{\odot}(.)\right)\right) \cdot \frac{\partial r\left(\vec{x}, \vec{\odot}(.)\right)}{\partial\left(\odot_x(.), \odot_y(.)\right)}, \tag{3.52}$$

with

$$\begin{aligned}\frac{\partial r\left(\vec{x}, \vec{\odot}(.)\right)}{\partial\left(\odot_x(.), \odot_y(.)\right)} &= \frac{\partial\left(I_r\left(\vec{\odot}(.)\right) - f(\vec{x})\right)}{\partial\left(\odot_x(.), \odot_y(.)\right)} = \begin{pmatrix} \frac{\partial I_r\left(\vec{\odot}(.)\right)}{\partial \odot_x(.)} \\ \frac{\partial I_r\left(\vec{\odot}(.)\right)}{\partial \odot_y(.)} \end{pmatrix}^T \\ &= \vec{\nabla} I_r^T\left(\vec{\odot}\left(\vec{W}, \vec{W}(\vec{x}, \Delta\vec{\theta})\right)\right),\end{aligned} \tag{3.53}$$

which is the image gradient of I_r evaluated at $\vec{\odot}\left(\vec{W}, \vec{W}(\vec{x}, \Delta\vec{\theta})\right)$.

Further we obtain

$$\begin{aligned}\mathbf{J}_{\vec{\odot}}\left(\vec{W}(\vec{x}, \Delta\vec{\theta})\right) &= \frac{\partial\left(\odot_x(.), \odot_y(.)\right)}{\partial\left(W_x(\vec{x}, \Delta\vec{\theta}), W_y(\vec{x}, \Delta\vec{\theta})\right)} \\ &= \begin{pmatrix} \frac{\partial \odot_x\left(\vec{W}, \vec{W}(\vec{x}, \Delta\vec{\theta})\right)}{\partial W_x(\vec{x}, \Delta\vec{\theta})} & \frac{\partial \odot_x\left(\vec{W}, \vec{W}(\vec{x}, \Delta\vec{\theta})\right)}{\partial W_y(\vec{x}, \Delta\vec{\theta})} \\ \frac{\partial \odot_y\left(\vec{W}, \vec{W}(\vec{x}, \Delta\vec{\theta})\right)}{\partial W_x(\vec{x}, \Delta\vec{\theta})} & \frac{\partial \odot_y\left(\vec{W}, \vec{W}(\vec{x}, \Delta\vec{\theta})\right)}{\partial W_y(\vec{x}, \Delta\vec{\theta})} \end{pmatrix}\end{aligned} \tag{3.54}$$

and

$$\begin{aligned}\mathbf{J}_{\vec{W}}(\Delta\vec{\theta}) &= \frac{\partial\left(W_x(\vec{x}, \Delta\vec{\theta}), W_y(\vec{x}, \Delta\vec{\theta})\right)}{\partial\left(\Delta a, \Delta b\right)} \\ &= \begin{pmatrix} \frac{\partial W_x(\vec{x}, \Delta\vec{\theta})}{\partial \Delta a} & \frac{\partial W_x(\vec{x}, \Delta\vec{\theta})}{\partial \Delta b} \\ \frac{\partial W_y(\vec{x}, \Delta\vec{\theta})}{\partial \Delta a} & \frac{\partial W_y(\vec{x}, \Delta\vec{\theta})}{\partial \Delta b} \end{pmatrix}.\end{aligned} \tag{3.55}$$

Due to the compact parameterization, this simplifies to

$$\mathbf{J}_{\vec{\odot}}\left(\vec{W}(\vec{x}, \Delta\vec{\theta})\right) \mathbf{J}_{\vec{W}}(\Delta\vec{\theta}) = \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} -\bar{y} & -1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \bar{y} & 1 \\ 0 & 0 \end{pmatrix}. \tag{3.56}$$

Based on these results we can assemble the auxiliary variable

$$\mathbf{J}^*(\vec{x}, \vec{W}, \Delta\vec{\theta}) = \frac{\partial r\left(\vec{x}, \vec{\odot}(.)\right)}{\partial\left(\odot_x(.), \odot_y(.)\right)} \mathbf{J}_{\vec{\odot}}\left(\vec{W}(\vec{x}, \Delta\vec{\theta})\right) \mathbf{J}_{\vec{W}}(\Delta\vec{\theta}) \tag{3.57}$$

to compactly write the Hessian as

$$\begin{aligned}
&\mathbf{A}\left(\vec{\odot}\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right)\right)\\
&\quad=\sum_{\vec{x}\in\Omega}\rho''\left(r(\vec{x},\vec{\odot}(.))\right)\cdot\mathbf{J}^{*\,T}(\vec{x},\vec{W},\Delta\vec{\theta})\mathbf{J}^{*}(\vec{x},\vec{W},\Delta\vec{\theta})\\
&\quad+\sum_{\vec{x}\in\Omega}\rho'\left(r(\vec{x},\vec{\odot}(.))\right)\cdot\frac{\partial\left(\mathbf{J}^{*}_{\Delta a}(\vec{x},\vec{W},\Delta\vec{\theta}),\mathbf{J}^{*}_{\Delta b}(\vec{x},\vec{W},\Delta\vec{\theta})\right)}{\partial(\Delta a,\Delta b)}.
\end{aligned}\tag{3.58}$$

3.3.3.2 *Inverse Warp Composition*

The key ingredient leading to the efficiency of the inverse compositional approach is the combination of warp compositions with a switch of the roles of the reference image f and the target image I_r. Consequently, the goal in each iteration is to take a step that minimizes

$$\begin{aligned}
F\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right)&=\sum_{\vec{x}\in\Omega}\rho\left(r\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right)\right)\\
&=\sum_{\vec{x}\in\Omega}\rho\left(I_r\left(\vec{W}\right)-f\left(\vec{W}(\vec{x},\Delta\vec{\theta})\right)\right).
\end{aligned}\tag{3.59}$$

and perform an *inverse* warp update according to

$$\vec{W}\leftarrow\vec{\odot}\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})^{-1}\right),\tag{3.60}$$

where

$$\vec{\odot}\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})^{-1}\right)=\begin{pmatrix}x+(\Delta a\bar{y}+\Delta b)-(a\bar{y}+b)\\ y\end{pmatrix}.\tag{3.61}$$

Note that to perform a derivation following Sect. 2.1, the Taylor expansion with respect to $\Delta\vec{\theta}$ has to be applied before the composition of the current warp with the inverse of the incremental update.

The cost function gradient and hence the Jacobian $\mathbf{J}_{F\circ\vec{W}}$ with respect to the parameter update is then

$$\begin{aligned}
\vec{g}^{\,T}\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right)&=\mathbf{J}_{F\circ\vec{W}}\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right)\\
&=\mathbf{J}_{F}\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right)\mathbf{J}_{\vec{W}}(\Delta\vec{\theta}),
\end{aligned}\tag{3.62}$$

where $\mathbf{J}_F$ is the Jacobian of the cost function with respect to the incremental warp update and $\mathbf{J}_{\vec{W}}$ is the Jacobian of the image warp with respect to the parameter update $\Delta\vec{\theta}$.

Here, the Jacobian $\mathbf{J}_F$ can be expressed as

$$\begin{aligned}\mathbf{J}_F\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right) = \sum_{\vec{x}\in\Omega} \rho'\left(r\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right)\right) \\ \cdot \frac{\partial r\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right)}{\partial\left(W_x(\vec{x},\Delta\vec{\theta}),W_y(\vec{x},\Delta\vec{\theta})\right)},\end{aligned} \tag{3.63}$$

with

$$\begin{aligned}\frac{\partial r\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right)}{\partial\left(W_x(\vec{x},\Delta\vec{\theta}),W_y(\vec{x},\Delta\vec{\theta})\right)} &= \frac{\partial\left(I_r\left(\vec{W}\right) - f(\vec{W}(\vec{x},\Delta\vec{\theta}))\right)}{\partial\left(W_x(\vec{x},\Delta\vec{\theta}),W_y(\vec{x},\Delta\vec{\theta})\right)} \\ &= -\begin{pmatrix}\frac{\partial f\left(\vec{W}(\vec{x},\Delta\vec{\theta})\right)}{\partial W_x(\vec{x},\Delta\vec{\theta})} \\ \frac{\partial f\left(\vec{W}(\vec{x},\Delta\vec{\theta})\right)}{\partial W_y(\vec{x},\Delta\vec{\theta})}\end{pmatrix}^T \\ &= -\vec{\nabla} f^T\left(\vec{W}(\vec{x},\Delta\vec{\theta})\right).\end{aligned} \tag{3.64}$$

Note that at $\Delta\vec{\theta} = \vec{0}$, only the gradient of the reference image at the original pixel position $\vec{x}$ has to be evaluated. The Jacobian of the warp is expressed as in (3.55).

Finally, the full Hessian is

$$\begin{aligned}&\mathbf{A}\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right) \\ &= \sum_{\vec{x}\in\Omega} \rho''\left(r\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right)\right) \cdot \mathbf{J}^{*\,T}(\vec{x},\vec{W},\Delta\vec{\theta})\mathbf{J}^*(\vec{x},\vec{W},\Delta\vec{\theta}) \\ &+ \sum_{\vec{x}\in\Omega} \rho'\left(r\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right)\right) \cdot \frac{\partial\left(\mathbf{J}^*_{\Delta a}(.),\mathbf{J}^*_{\Delta b}(.)\right)}{\partial(\Delta a,\Delta b)},\end{aligned} \tag{3.65}$$

using the auxiliary variable

$$\mathbf{J}^*(\vec{x},\vec{W},\Delta\vec{\theta}) = \frac{\partial r\left(\vec{W},\vec{W}(\vec{x},\Delta\vec{\theta})\right)}{\partial\left(W_x(\vec{x},\Delta\vec{\theta}),W_y(\vec{x},\Delta\vec{\theta})\right)}\mathbf{J}_{\vec{W}}(\Delta\vec{\theta}). \tag{3.66}$$

Dropping all second-order derivatives, we obtain the approximate Gauss-Newton Hessian at $\Delta\vec{\theta} = \vec{0}$:

$$\begin{aligned}\mathbf{A}_{GN}(\vec{W}) &= \sum_{\vec{x}\in\Omega} \rho''\left(r\left(\vec{W},\vec{x}\right)\right) \cdot \mathbf{J}^{*\,T}(\vec{x},\vec{W},\vec{0})\mathbf{J}^*(\vec{x},\vec{W},\vec{0}) \\ &= \sum_{\vec{x}\in\Omega} \rho''\left(r\left(\vec{W},\vec{x}\right)\right) \cdot \begin{pmatrix}\bar{y}^2\left(\frac{\partial f(\vec{x})}{\partial x}\right)^2 & \bar{y}\left(\frac{\partial f(\vec{x})}{\partial x}\right)^2 \\ \bar{y}\left(\frac{\partial f(\vec{x})}{\partial x}\right)^2 & \left(\frac{\partial f(\vec{x})}{\partial x}\right)^2\end{pmatrix}\end{aligned} \tag{3.67}$$

and consequently the approximate Levenberg-Marquardt Hessian

$$\mathbf{A}_{LM}(\vec{W}) = \mathbf{A}_{GN}(\vec{W}) + \gamma \mathsf{diag}\left\{\mathbf{A}_{GN}(\vec{W})\right\}. \tag{3.68}$$

Under the assumption that the warp parameters are close to the correct solution, the equivalence to first order in $\Delta\vec{\theta}$ of the inverse compositional, forwards compositional, and the original optimization algorithm of Sect. 3.2.4 can be shown. For details we refer to [Baker and Matthews, 2004].

3.3.4 *Remarks and Implementation Details*

3.3.4.1 *Precomputation*

Taking a closer look at the components of the approximate Hessian in (3.67), it can be seen that as a result of the inverse compositional formulation almost all terms are now in fact independent of the current parameter vector estimate. However, two details have to still be considered: The estimation of the true image signal f as proposed in Sect. 3.2.4.4 for the PHT method has to be omitted, directly using the left image sample as the reference estimate. Further, for the function ρ a simple loss, such as the squared loss, has to be used. Robust loss functions which perform a residual-dependent scaling, such as the Huber loss, would require the Hessian to still be recomputed in every iteration due to the term $\rho''(.)$.

Finally, with (3.67) we obtain a formulation of the FPHT optimization procedure where the approximate Hessian needs to be precomputed only once and remains constant over all subsequent iterations. Furthermore, all necessary image gradients are evaluated only once on the left reference image and do not have to be warped during optimization.

Note that the inverse compositional approach cannot be readily applied to the PHT method since the required identity warp can only be achieved by a fronto-parallel plane at infinity with the used 3D plane representation . An alternative is the inverse additive algorithm as proposed by Hager and Belhumeur [1998], switching the image roles by a similar change in variables. However the naive version of the inverse additive algorithm does not result in a notable gain in efficiency, and the PHT 3D plane parameterization is not applicable to the efficient special cases described in [Hager and Belhumeur, 1998].

3.3.4.2 *Parallelization*

Since both PHT and FPHT operate on isolated, independent image patches, hypothesis testing across all patches can be done in parallel. Even the hypothesis models of the obstacle and free-space hypotheses of a single patch can be optimized independently from each other. Therefore, the core of the proposed detection algorithms benefits

from massive parallelization, using either multi-core CPUs, GPUs, or dedicated hardware such as FPGAs.

3.3.4.3 *Subsampling*

In order to further increase efficiency by reducing redundant computations, we place the patches to be tested on subsampled image positions. By default we employ subsampling of stride $s_x = s_y = 2$, which means that only every other image point in both horizontal and vertical direction is actively being tested. Note that each patch still makes use of the full underlying image data to perform the hypothesis test. Our experiments show that a stride of two provides a significant boost in efficiency, while virtually no detection performance is being sacrificed.

3.4 OBJECT REPRESENTATION AND TRACKING

3.4.1 *Mid-Level Representation*

Inspired by the compactness and flexibility of the Stixel representation described in Sect. 3.1.3, we propose a corresponding extension for point-wise object detection approaches such as PHT, FPHT and also the PC method of [Manduchi et al., 2005, Broggi et al., 2011]. The aim is to create a mid-level representation similar to the Stixel algorithm, reducing the amount of output data and at the same time increasing robustness. Furthermore, the flexibility of this representation is particularly beneficial for handling arbitrarily shaped objects in complex and unstructured scenes, such as regularly encountered in inner-city traffic.

The proposed Cluster-Stixels (CStix) approach does not perform optimization along image columns like the actual Stixel algorithm, but consists of a 3D clustering and a splitting step instead (see Alg. 3.1 and [Pinggera et al., 2016]). The resulting representation exhibits characteristics very similar to the traditional *Stixel World* of Pfeiffer and Franke [2011], with the only difference that here Stixels are not guaranteed to be vertically aligned and may overlap in the image plane. Fig. 3.6 illustrates the generation of the Cluster-Stixels representation from the obstacle detection output shown in Fig. 3.4. A different example of the CStix output based on FPHT point detections is depicted in Fig. 3.7. The individual processing steps are described in the following.

3.4.1.1 *Clustering*

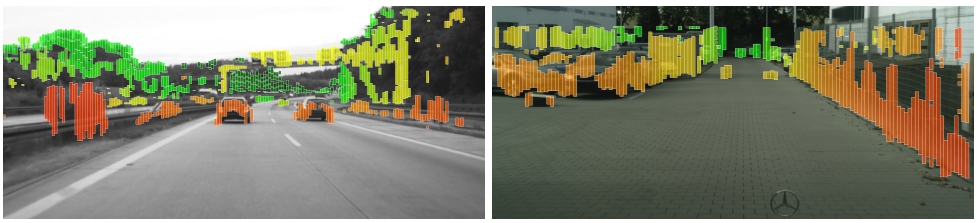
In the first step, density-based geometric point clustering in 3D space is performed via a modified DBSCAN algorithm [Ester et al., 1996]. The DBSCAN algorithm considers the spatial neighborhoods of all data samples and groups together points which are closely packed, while treating sparse points in low-density regions as outliers. It handles arbitrarily shaped clusters and does not require an initial estimate of the total number of clusters.

We approximate the spherical point neighborhood regions of the original algorithm by cuboids for efficient data access using bulk-loaded R-trees [Guttman, 1984, Leutenegger et al., 1997]. Furthermore, we introduce several suitable modifications to take the characteristics of point clouds resulting from stereo algorithms in general - and from the detection algorithms presented above in particular - into account.

First, the dimensions of neighborhood regions are scaled with the points' absolute distances from the camera. This is done according to

(a) Density-based clustering of obstacle points (random colored)

(b) Cluster splitting (random colored)



(c) CStix result, color-coded by distance

Figure 3.6: Generation of the Cluster-Stixels (CStix) representation from point-wise obstacle detections.

the estimated disparity noise σ_d and a potential stride s_x, s_y between points resulting from pixel subsampling:

$$L_R = \left(\frac{f_x B \cdot Z_C}{f_x B - Z_C \cdot \sigma_d} + \epsilon_L\right) - \left(\frac{f_x B \cdot Z_C}{f_x B + Z_C \cdot \sigma_d} - \epsilon_L\right), \tag{3.69}$$

$$W_R = 2 \cdot \left(\epsilon_W + \frac{Z_C}{f_x} \cdot s_x\right), \tag{3.70}$$

$$H_R = 2 \cdot \left(\epsilon_H + \frac{Z_C}{f_y} \cdot s_y\right). \tag{3.71}$$

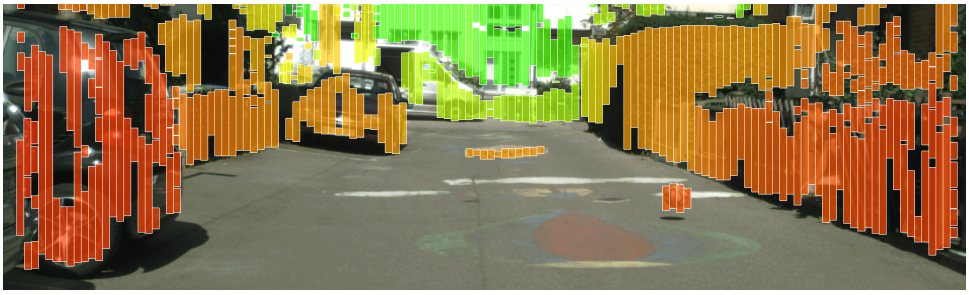
Here, ϵ_L, ϵ_W, ϵ_H represent the original parameters defining half the region size, L_R, W_R, H_R represent the adapted full dimensions, Z_C denotes the point position along the principal axis and f_x, f_y and B the focal length and the baseline length of the camera.

Also, the parameter defining the minimum number of cluster points is scaled with the distance from the camera, where the scaling formula is very similar to the coordinate scaling used in [Nedevschi et al., 2004a]:

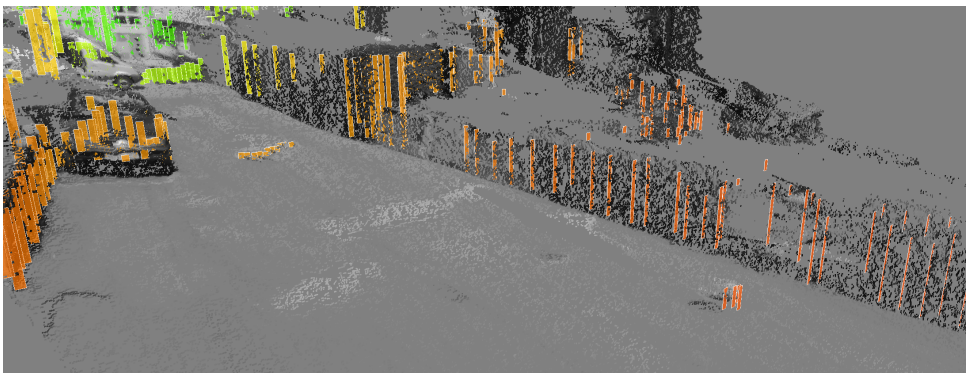
$$minPts = minPts_0 + k \cdot \frac{f_x}{Z_C}, \tag{3.72}$$

(a) Left stereo input image

(b) FPHT obstacle point detections



(c) Output of the FPHT-CStix method



(d) 3D view of the CStix representation

Figure 3.7: Exemplary output of the FPHT method in an urban scenario with challenging obstacles, illustrating the raw obstacle point representation as well as the corresponding mid-level Stixel representation CStix in the image plane and in 3D. The FPHT results are color-coded by distance, the dense underlying point cloud in the 3D view represents the SGM output used for initialization.

with a free parameter k allowing for manual control of the scaling magnitude.

Finally, the horizontal orientation of the rectangular neighborhood regions is aligned to the viewing rays of the camera.

The adaptive DBSCAN algorithm allows for the use of meaningful clustering parameters, combining real-world dimensions and disparity uncertainty, and avoids discretization artifacts typical of e.g. scaled grid maps.

Note that for the PC approach the clustering step can be omitted, since the detection algorithm itself already provides a set of meaningful clusters (see Sect. 3.1.2).

3.4.1.2 *Splitting*

After the clustering phase, each cluster is split vertically in the image domain, yielding a set of Stixel-like vertical boxes. The vertical splitting step strictly enforces a fixed box width to ensure the characteristic Stixel layout seen in Fig. 3.2.

Optionally, an additional horizontal splitting step may be performed to counter occasional cases of under-segmentation. This step performs recursive splits only as long as the disparity variance within a Stixel box exceeds a certain threshold. To analyze the disparity variance, we employ a precomputed disparity map which is also used to initialize the PHT and FPHT approaches.

3.4.2 *Object Representation*

As an alternative to the mid-level Stixel output representation, we also consider the classical Bounding Box (BB) object description. It provides an intuitive and well established way of representing compact objects in structured environments. Based on this representation, individual objects may be tracked over time for improved consistency and robustness (see Sect. 3.4.3).

To generate the bounding box representation from the raw object points, the same clustering step as for the Stixel output is applied. Subsequently, a robust box fit is computed for each cluster either in 3D space or after projection into the 2D image domain. For representing objects at medium or long ranges, it is adequate to fit a 2D box in the image domain. The full 3D box is used only if at least two sides of the object are actually observable, i.e. if each side projects to a minimum number of image pixels. This is usually the case only for very large objects or close range detections.

To compute the robust fit, a certain fraction of all points in the cluster are considered as potential outliers along each dimension. The box size and position are then determined by the inlier points. Optimal estimation of the object position, in particular the object distance, is analyzed in detail in Chapter 4.

Algorithm 3.1 Mid-level representation: Cluster-Stixels (CStix)

Input

- list of 3D obstacle points $\vec{P} = \left(\vec{X}_1 \ldots \vec{X}_n\right)$, e. g. from PHT, FPHT, PC
- dense or sparse disparity map $\mathcal{D}$

Output

- list of obstacle Cluster-Stixels $C\vec{S}tix$

Algorithm

function MIDLEVELREP($\mathcal{D}$)
  $\vec{Cl} \leftarrow$ ADAPTIVEDBSCAN($\vec{P}$) ▷ Compute list of obstacle clusters $\vec{Cl}$ with associated points
  $C\vec{S}tix \leftarrow$ SPLITANDFIT($\vec{Cl}$, $\mathcal{D}$) ▷ Split clusters $\vec{Cl}$ and fit Bounding Boxes (BB)
  return $\{C\vec{S}tix\}$
end function

function SPLITANDFIT($\vec{Cl}$, $\mathcal{D}$)
  for all $Cl \in \vec{Cl}$ **do**
    $C\vec{S}tix \xleftarrow{+}$ SPLITVERTICALLY(Cl, $width$) ▷ split vertically and fit BB with fixed Stixel width
    $C\vec{S}tix \xleftarrow{+}$ SPLITHORIZONTALLY($C\vec{S}tix$, Cl, $\mathcal{D}$) ▷ split horizontally until disparity variance in BB is below threshold
  end for
  return $\{C\vec{S}tix\}$
end function

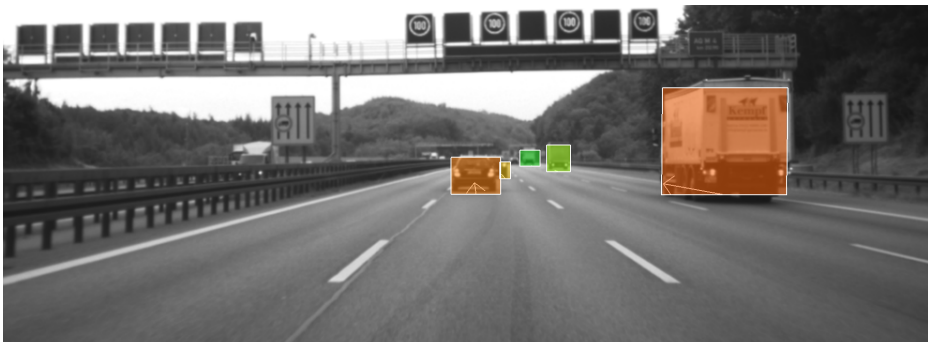
Again, as an additional measure to prevent under-segmentation errors, cluster splitting steps may be introduced based on an analysis of the disparity distribution within each box. An example of the obtained bounding box output based on FPHT point detections is illustrated in Fig. 3.8.

3.4.3 *Object Tracking*

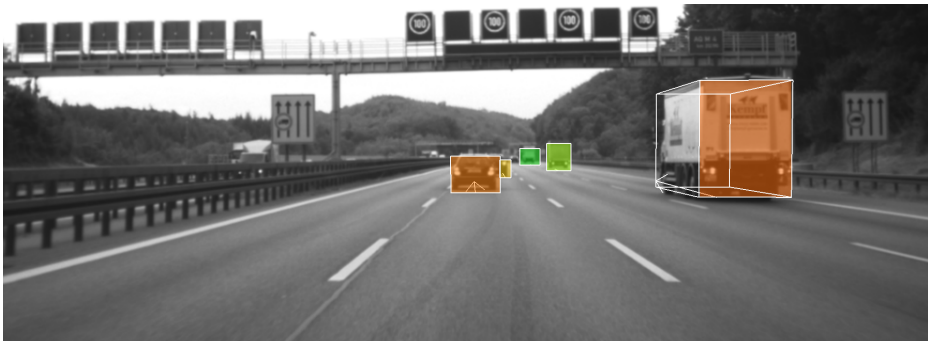
To illustrate the potential of embedding the proposed detection methods within a temporal filtering framework, we implement a multi-object tracking system as follows. The system is based on the output of the bounding box object representation described above and utilizes a straightforward nearest-neighbor extended Kalman filter approach [Pulford, 2005]. It is best suited for scenarios involving relatively compact objects with regular motion patterns, such as vehicles moving in highway traffic. More elaborate approaches out of the vast field of multi-object tracking research could be applied (see e.g. [Pulford, 2005, Granström et al., 2017]), but this lies outside the scope of the present work. For a more detailed derivation and analysis of the system and measurement models described in the following we refer to [Rabe, 2011].

(a) Left stereo input image

(b) FPHT obstacle point detections



(c) Output of the FPHT-BB (2D) representation



(d) Output of the FPHT-BB (3D) representation

Figure 3.8: Exemplary output of the FPHT method in a highway scenario, illustrating the raw obstacle point representation as well as the corresponding tracked bounding box object representations. The shown results are color-coded by distance, white arrows indicate the velocity estimated by the tracking algorithm. Note that here the region of interest is restricted to the road area ahead.

3.4.3.1 *System model*

For simplicity, tracked objects are reduced to point targets located at the center of the bounding box representation. The state of each object is then described by the vector $\vec{\mathcal{X}} = (X, Y, Z, X', Y', Z')^T$, holding the position in 3D space and the respective velocity components. The discrete-time system model describing the linear motion of an object in the coordinate system of the moving observer, i.e. the ego-vehicle, at time step k is written as

$$\vec{\mathcal{X}}_k = \begin{pmatrix} \mathbf{R}_k & \mathbf{0}_{(3\times3)} \\ \mathbf{0}_{(3\times3)} & \mathbf{R}_k \end{pmatrix} \mathbf{A}_{k|w} \vec{\mathcal{X}}_{k-1} + \begin{pmatrix} \vec{t}_k \\ \vec{0}_{(3)} \end{pmatrix} + \vec{\omega}_k, \tag{3.73}$$

where $\mathbf{A}_{k|w} = \begin{pmatrix} \mathbf{I}_{(3\times3)} & \Delta t \mathbf{I}_{(3\times3)} \\ \mathbf{0}_{(3\times3)} & \mathbf{I}_{(3\times3)} \end{pmatrix}$ is the state transition matrix of the model in the world coordinate system. $\mathbf{R}_k$ and $\vec{t}_k$ describe the motion of the ego-vehicle, which in this case is obtained from vehicle odometry. $\mathbf{Q}_k$ is the covariance matrix of the system model, with $\vec{\omega}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$.

Additionally, a separate exponential smoothing filter is used to process the estimates of the object bounding box dimensions.

3.4.3.2 *Measurement model*

The position $\vec{X} = (X, Y, Z)^T$ of an object in 3D space is projected to $\vec{x} = (x, y)^T$ in the reference image, with the corresponding stereo disparity d. Using the extended stereo projection matrix $\grave{\mathbf{P}}$ as defined in (2.45), this can be written as

$$\begin{pmatrix} x \\ y \\ d \end{pmatrix} = \begin{pmatrix} \mathbf{I}_{(3\times3)} & \vec{0}_{(3)} \end{pmatrix} \frac{1}{w} \grave{\mathbf{P}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = h(\vec{\mathcal{X}}), \tag{3.74}$$

where w denotes the homogeneous normalization factor. Considering the additive measurement noise vector $\vec{v}_k$ we arrive at the non-linear measurement model

$$\vec{z}_k = h\left(\vec{\mathcal{X}}_k, \vec{v}_k\right). \tag{3.75}$$

The measurement vector $(x, y, d)^T$ is obtained directly from the output of the object detection algorithm. Here it is crucial to compute a high-accuracy estimate of the disparity d, as it has a major impact on the estimated distance and velocity of observed objects in 3D space. The problem of finding optimal object disparity estimates is discussed in detail in Chapter 4.

3.4.3.3 *Data association*

To associate object detections and thus measurements with existing tracks, a straightforward nearest-neighbor scheme with gating windows is applied. Only a single detection may be associated with any given track, and only if all of the following conditions are true:

- The detection lies within a given gating window around the predicted track position, specified in 3D space.
- The detection lies within a given gating window around the projected track position, specified in the 2D image domain. Projection from the last time step $k-1$ is performed via sparse optical flow [Tomasi and Kanade, 1991].
- The detection is the nearest neighbor of the track when considering all detections lying within the specified gating windows.

New tracks are created from all unassociated detections, however, a minimum existence duration or age is required before a track is added to the system output.

3.5 EVALUATION

In the following sections we present a thorough performance analysis and evaluation of the proposed approaches in several challenging real-world application scenarios. We consider the primary detection algorithms PHT and FPHT as well as the described mid-level and object-level extensions. The evaluation methodology is mainly based on the work presented in [Pinggera et al., 2016], but also includes some extensions of [Ramos et al., 2017].

First, the various performance metrics and the dedicated datasets used in the evaluation are introduced. Second, the baselines and the evaluation methodology are described. Finally, the in-depth qualitative and quantitative analysis of the results is presented.

3.5.1 *Evaluation Metrics*

To quantitatively analyze the detection performance of the different approaches, we define pixel-level, instance-level and object-level metrics derived from related computer vision problems, always keeping the application focus in mind.

3.5.1.1 *Pixel-Level Metrics*

As a first metric we define a Receiver-Operator-Characteristic (ROC) curve that compares the Pixel-Level True Positive Rate ($\mathrm{TPR_p}$) over the Pixel-Level False Positive Rate ($\mathrm{FPR_p}$). This ROC curve is generated by performing a parameter sweep and computing the convex hull over the

results of all evaluated parameter configurations. TPR_p and FPR_p are computed as

$$TPR_p = \frac{TP_p \cdot k_1^2 \cdot k_2^2}{GT_o}, \tag{3.76}$$

$$FPR_p = \frac{FP_p \cdot k_1^2 \cdot k_2^2}{GT_f}. \tag{3.77}$$

Here TP_p and FP_p refer to the number of true and false pixel-wise predictions that a given method produces, which are evaluate with respect to the annotated image areas. k_1 and k_2 are scaling factors that compensate for the downsampling and subsampling settings of some of the evaluated methods. Finally, GT_o and GT_f represent the total number of ground truth pixels labeled as obstacle or free-space, respectively.

3.5.1.2 *Instance-Level Metrics*

The main drawback of the above described ROC curve is its bias towards object instances that cover large image areas. Therefore, we apply a second metric on the pixel level which overcomes this disadvantage. It analyzes the average Instance-Level Intersection (iInt) between the algorithm output predictions and the pixel-wise ground truth annotations. This is inspired by an instance-level variation of the Jaccard Index, known as Instance-Level Intersection over Union (iIoU) (see [Cordts et al., 2014, 2016]. The Instance-Level Intersection (iInt) result is computed as

$$iInt = \frac{\sum_i \frac{TP_p^i}{TP_p^i + FN_p^i}}{|O|_{GT}}, \tag{3.78}$$

where TP_p^i and FN_p^i represent pixel-wise true positives and false negatives for a particular object instance and $|O|_{GT}$ represents the total number of ground truth objects. In this way, the imbalance caused by absolute object sizes is effectively eliminated.

3.5.1.3 *Object-Level Metrics*

The pixel- and instance-level metrics yield important insights regarding algorithm behavior, which can be utilized to determine suitable working points for the most important system parameters as in [Pinggera et al., 2015, 2016]. However, from an application perspective, more valuable performance figures are obtained by considering object-level metrics. Therefore, we additionally analyze the overall object-level true positive rate, also referred to as Detection Rate (DR). It represents the fraction of ground-truth objects which are detected successfully by the system (see [Ramos et al., 2017]).

Depending on the considered output representation of the algorithm, in the following evaluation a ground-truth object is regarded as being detected correctly if it is represented by at least one true positive Stixel or bounding box. Here, a true positive is defined by an overlap of at least 50% of its image area with the ground-truth object. By requiring only a single corresponding true positive per object, the resulting detection rate represents an optimistic estimate with regard to the overall application performance.

3.5.1.4 *False Positives*

In order to thoroughly analyze algorithm behavior and detection performance, a suitable definition of False Positives (FPs) is required. For the raw object point output under the pixel-level metric of Sect. 3.5.1.1, false positives simply correspond to the incorrectly detected pixels.

With regard to the instance-level and object-level metrics, we define false positives as follows: A Stixel or bounding box is defined as a false positive if the overlap with the labeled free-space region is larger than 50% of its area. For the object-level evaluation, we additionally define a tolerance region of 0.25° (approx. 10 px) around ground-truth object borders, since errors caused by border artifacts such as foreground fattening are considered acceptable for this task and should not influence the metric.

In the following, either the total number of false positives or the number of false positives per frame is considered.

3.5.2 *Datasets*

3.5.2.1 Lost and Found *Dataset*

The first dataset used for evaluation is the *Lost and Found* dataset presented in [Pinggera et al., 2016]. It consists of recordings from 13 different challenging urban street scenarios, featuring 37 different obstacle types. The selected scenarios contain particular challenges including irregular road profiles, long object distances, different road surface appearance as well as illumination changes. The objects to be detected are selected as a representative set of generic, small obstacles that may actually appear on the road in practice (see Fig. 3.9). These objects vary in size and material, which are factors that define how hazardous an object may be for a self-driving vehicle in case the obstacle is placed within the driving corridor. Very flat objects (i.e. lower than 5 cm) are treated as non-hazardous and thus are not taken into account in the results reported in Sect. 3.5.5.1.

The *Lost and Found* dataset consist of a total of 112 video stereo sequences with coarse annotations of free-space areas and fine-grained annotations of the obstacles on the road. Annotations are provided for every tenth frame, giving a total of 2104 annotated images. Each object is labeled with a unique ID, allowing for an instance-level analysis. An ex-

Figure 3.9: Overview of objects included in the *Lost and Found* dataset.

Table 3.1: Details on the *Lost and Found* dataset subsets. Numbers in parentheses represent unseen test items not included in the training set.

Subset	Sequences	Frames	Locations	Objects
Train/Val	51	1036	8	28
Test	61	1068	5 (5)	35 (9)

ample image and the corresponding ground truth annotation is shown in Fig. 3.10.

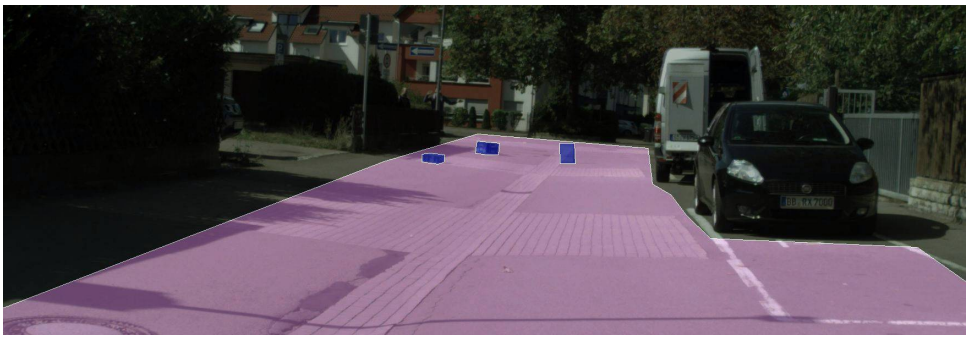
The dataset is split into a Train/Validation subset and a Test subset. Each of these subsets consists of recordings taken in completely different surroundings, covering a similar number of video sequences, frames and objects (see Table 3.1). The Test subset contains nine previously unseen objects that are not present in the Training/Validation subset. Further, the test scenarios can be considered to be more difficult than the training scenarios, amongst others due to more complex road profile geometries.

The stereo camera setup features a baseline of 21 cm and a focal length of 2300 pixels, with spatial and radiometric resolutions of 2048×1024 pixels and 12 bits. While the dataset consists of full color images, the methods developed in this work are applied only to grayscale data.

3.5.2.2 Highway Detection *Dataset*

The second dataset, initially presented in [Cordts et al., 2014], was utilized in [Pinggera et al., 2015] for evaluating the PHT method within the context of long range object detection in highway scenarios. It consists of 2000 frames of manually labeled stereo images, taken by a test vehicle in highway traffic. Relevant obstacles are represented by other traffic participants. Non-occluded vehicles up to distances of 300 m are labeled with pixel-accuracy in every frame, every tenth frame also includes a pixel-wise free-space labeling. The used stereo camera features a baseline of 39 cm, a focal length of 1260 pixels and provides grayscale images with spatial and radiometric resolutions of 1024×440 pixels and 12 bits.

(a) Left stereo input image



(b) Pixel-wise ground truth annotation

Figure 3.10: Example image from the *Lost and Found* dataset and corresponding ground truth annotation. Free-space is shown in purple, objects are marked in blue. This scene features three challenging obstacles of different heights positioned in a suburban area. Also note the slight lateral curvature in the road surface.

Compared to the *Lost and Found* dataset, the data of this collection is characterized by a more structured environment and a smoother road profile. However, it features moving objects at extremely long distances which have to be detected successfully in this application scenario.

3.5.3 *Baselines*

To assess the performance of the proposed detection approaches with respect to established reference algorithms, the following methods as introduced in Sect. 3.1 serve as baselines for evaluation:

- Stixels [Pfeiffer and Franke, 2011] (see Sect. 3.1.3).
- Point Compatibility (PC) [Manduchi et al., 2005, Broggi et al., 2011] (see Sect. 3.1.2).

This selected set of algorithms has been proven to perform very well in a large range of practical experiments, particularly in real-world, real-time, in-vehicle operation. Further, these methods are based on varying underlying detection concepts and provide different output representations,

(a) Left stereo input image

(b) Pixel-wise ground truth annotation

Figure 3.11: Example image from the *Highway Detection* dataset and corresponding ground truth annotation. Free-space is shown in purple, objects are marked in blue. This scene includes relevant objects at distances of up to 200 m.

allowing for a direct comparison using the various evaluation metrics defined above.

3.5.4 *Methodology*

All proposed and baseline methods as described above are included in our experiments for evaluation. For all point-based methods PHT, FPHT and PC, by default we employ subsampling of stride $s_x = s_y = 2$, which means that only every other image point in both horizontal and vertical direction is actively being tested. To further investigate the trade-off of efficiency and detection performance, we optionally scale down the input images by an additional factor of two for even faster execution. In the following, results including this input data reduction step will be denoted as *downsampled*.

To choose a suitable working point with regard to the principal parameters of each method, we perform a parameter sweep on the *Lost and Found* training set. The considered parameters are:

- PHT/FPHT: Patch size $h \times w$, minimum eigenvalue of the approximate Hessian λ_{min}, GLRT decision threshold γ.

- PC: Maximum angle $\check{\varphi}$, minimum height H_{min}, maximum connection height H_{max}.
- Stixels: Vertical cut costs used in dynamic programming.

The PHT and FPHT approaches are relatively robust to the exact choice of the plane normal bounds. Here we set $\check{\varphi}_f = 25°$ and $\check{\varphi}_o = 45°$. For PHT we reduce the number of free parameters in our evaluation by locking the plane normal component n_X at 0, which preserves sufficient flexibility for the considered scenarios. The plane parameter vectors are initialized using a dense disparity map, precomputed via Semi-Global Matching (SGM) as described in [Gehrig et al., 2015]. The same disparity map provides the input to the PC and Stixel algorithms. For both PHT and FPHT we use a quadratic loss function ρ. Interestingly, in our experiments we did not observe a noticeable performance penalty compared to using a robust function such as the Huber loss. In either case most gross errors are filtered out by the model consistency checks of Sect. 3.2.5.

For the computation of the mid-level Cluster-Stixels and object-level bounding box representation we use a fixed, manually optimized set of parameters. In fact, these exact parameter values were found to have a much lower impact on the final results than the parameters optimized in the sweep described above.

3.5.5 *Results*

3.5.5.1 Lost and Found *Dataset*

QUANTITATIVE RESULTS: PIXEL-LEVEL First, the primary methods (PHT, FPHT, PC and Stixels) are benchmarked using the *Lost and Found* training subset and the described pixel-level ROC curve. For the purpose of this first evaluation, we perform a parameter sweep as described above. The best performing parameter configurations are then determined by computing the convex hull over the True Positive Rate (TPR) and False Positive Rate (FPR) results (see Fig. 3.12). Note that the main purpose of this curve is method-specific parameter optimization. Direct comparison of the different curves has to be approached with care, as effects such as the large-object-size bias mentioned in Sect. 3.5.1.2 have to be considered.

Once the best performing parameter sets have been determined, a second pixel-level ROC curve is computed on the test subset, including the primary approaches along with their corresponding Cluster-Stixels extensions (Fig. 3.13).

The results show that the performance of all methods except Stixels is rather consistent across training and test subsets. Stixels perform notably worse on the test subset, even taking the FPR beyond the range shown in Fig. 3.13. The main reason for this effect are the challenging road profiles of the test set, where errors in the Stixels' road estimation module have fatal consequences with regard to the FPR.

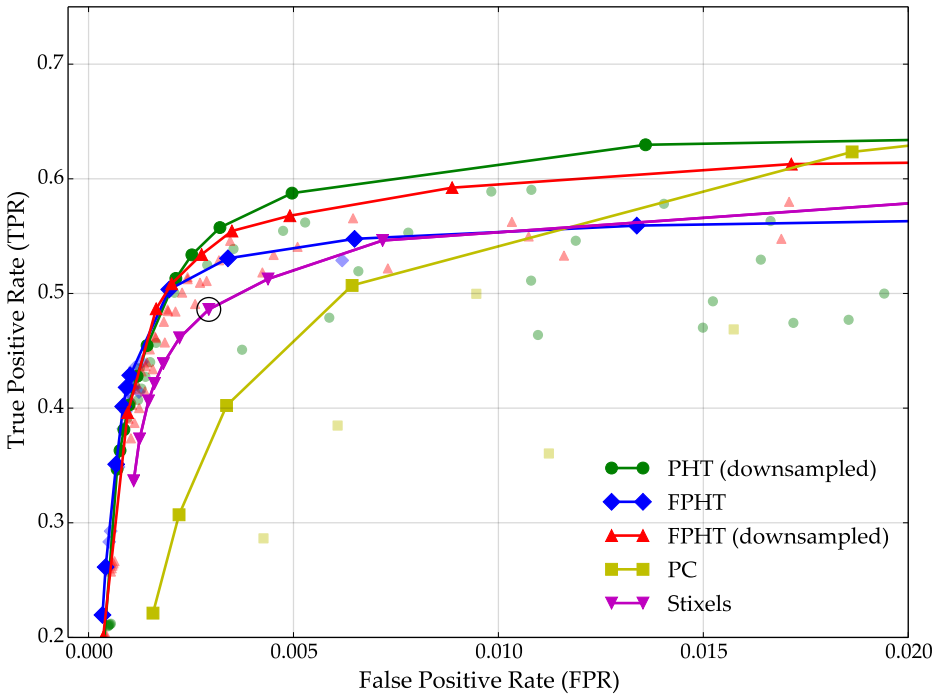


Figure 3.12: Pixel-level TPR over FPR (*Lost and Found* training subset). Solid curves represent the convex hulls of the respective parameter sweep results.

Fig. 3.13 further illustrates the gain that the proposed mid-level Cluster-Stixels representation provides over the raw obstacle point output of PHT, FPHT and PC. The Cluster-Stixels facilitate the propagation of detection results into object areas with low texture, while being compact and flexible enough to approximate the target object shapes without introducing a significant amount of false positive pixels. Additionally, sporadic false positive pixels are removed by the underlying clustering algorithm.

QUANTITATIVE RESULTS: INSTANCE-LEVEL Next, we compare the Cluster-Stixels output of the proposed and baseline approaches using the defined instance-level iInt metric in Fig. 3.14. The results clearly show that the PHT/FPHT approaches significantly outperform both baselines, yielding a relative improvement of 30% to 80% over PC at any given working point. Here also the negative impact of downsampling becomes visible, as it mainly affects the smallest object instances at long distances, which are now weighted equally by this metric. Notably, it can be seen that the FPHT method performs equal to or even slightly better than the PHT variant. This is most likely due to the more direct parameterization of FPHT and hence the simpler formulation of the underlying optimization problem.

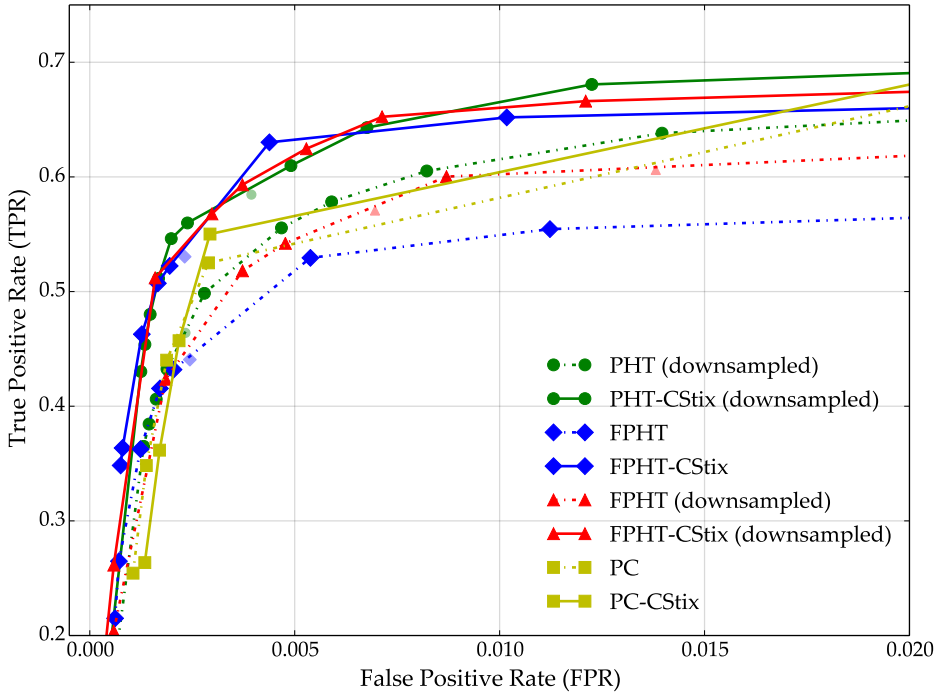


Figure 3.13: Pixel-level TPR over FPR (*Lost and Found* test subset).

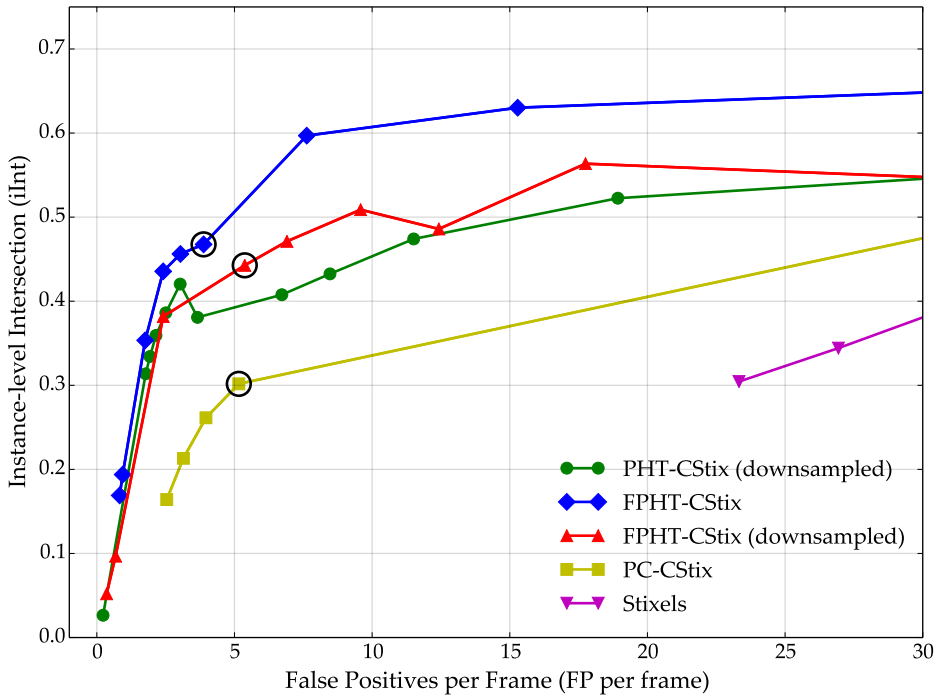


Figure 3.14: Instance-level iInt over FP per frame (*Lost and Found* test subset).

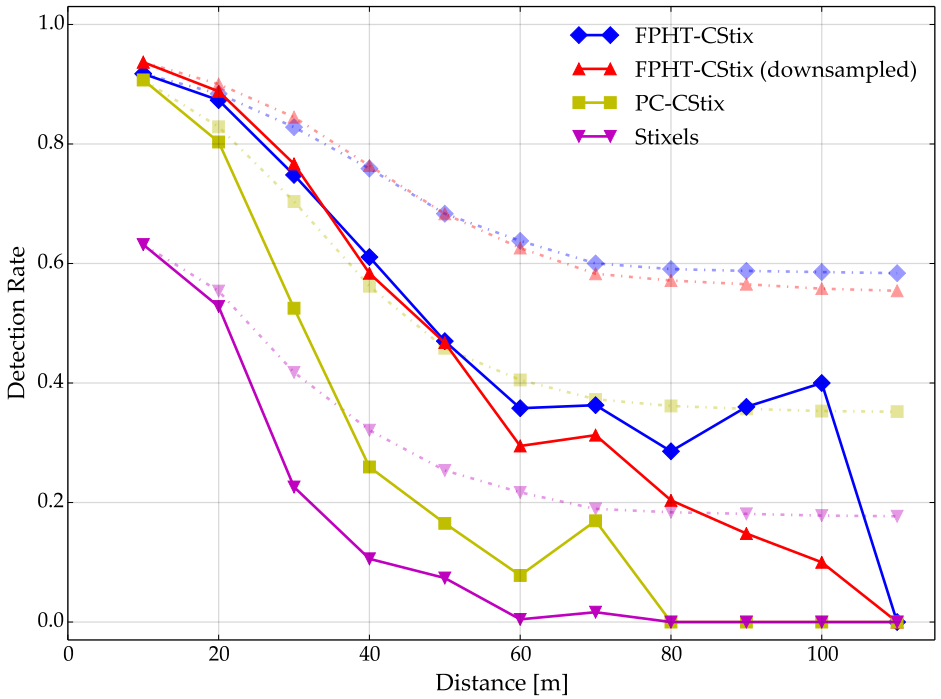


Figure 3.15: Object detection rates over object distance (*Lost and Found* test subset). Solid curves illustrate the detection rate at each single range bin while dashed curves represent the integrated detection rate up to a given distance.

QUANTITATIVE RESULTS: OBJECT-LEVEL Finally, we evaluate the object-level performance of FPHT-CStix and the baseline methods by considering the overall detection rates and false positives on the *Lost and Found* test set (see Table 3.2). Additionally, we analyze the detection rates as a function of object distance from the camera in Fig. 3.15. In order to arrive at overall detection and false positive scores for each approach, we select appropriate working points in the parameter space from the instance-level results of Fig. 3.14. An exception is made for the Stixel algorithm, where we select a suitable working point based on the training subset (Fig. 3.12). The respective points are marked accordingly in each plot.

Overall, Stixels achieve a detection rate of less than 18%, with massive false positive rates due to the variable ground profiles included in this challenging dataset. As illustrated in Fig. 3.15, objects beyond a distance of 60 m are not detected at all.

The PC-CStix approach yields approximately twice the detection rate of Stixels, with a much lower false positive rate due to its flexible detection criterion. Here, some objects are detected up to a distance of approximately 80 m.

The FPHT-CStix variants obtain by far the highest detection rates of just below 60%, with detections ranging up to a distance of 110 m. At the same time, false positive rates are significantly reduced compared

Table 3.2: Quantitative object-level results on the *Lost and Found* test subset.

	Detection Rate [%]	FP per frame	% frames with FP
Stixels [Pfeiffer and Franke, 2011]	17.7	41.62	60.0
PC-CStix [Manduchi et al., 2005]	35.2	1.50	45.6
FPHT-CStix (downsampled)	55.4	0.57	28.0
FPHT-CStix	58.4	0.29	15.1

to Stixels as well as PC-CStix. Owing to the chosen working points, the full-resolution version of FPHT-CStix produces only about half the false positives as the downsampled version, while still providing a larger detection range.

Nevertheless, from an application perspective even the lowest false positive numbers listed in Table 3.2 still appear rather high. However, it is worth noting that the dataset is specifically designed to highlight rare challenges, including different vertically curved road profiles and unusual road surface textures and drawings. Furthermore, in our experiments FPHT-CStix false positives appear approximately four times less frequently within the actual driving corridor than false positives in the outer, less relevant image parts.

QUALITATIVE RESULTS To complement the quantitative evaluation, Fig. 3.16 depicts qualitative results of the evaluated methods on three example scenarios from the *Lost and Found* test subset. The left column shows a typical example of a small road hazard, a bobby car, in a residential area. In this case, due to the flat road profile and the medium object size, all methods are able to successfully detect the object.

In the middle column, an example with objects at large distances on a bumpy surface is shown. At such distances, the signal-to-noise ratio of the disparity measurements drops significantly, leading to a very low quality of the constructed 3D point cloud. Thus, neither the Stixel nor the PC approaches are able to detect the relevant objects in the scene. In contrast, the FPHT methods, which operate on the image data directly, still perform reasonably well at such large distances.

The scene in the rightmost column illustrates a rather challenging case for geometry-based obstacle detection approaches. A noticeable double kink in the longitudinal road profile would require an extremely accurate road model estimation for the Stixel method to be able to detect such small objects. While the PC and FPHT methods are invariant to such conditions, only FPHT succeeds in actually detecting the tire on the left side of the image. The tire simply appears to be not prominent enough for a PC-based detection. Considering the FPHT-CStix results, it can be seen that the detections cover larger portions of the obstacle than

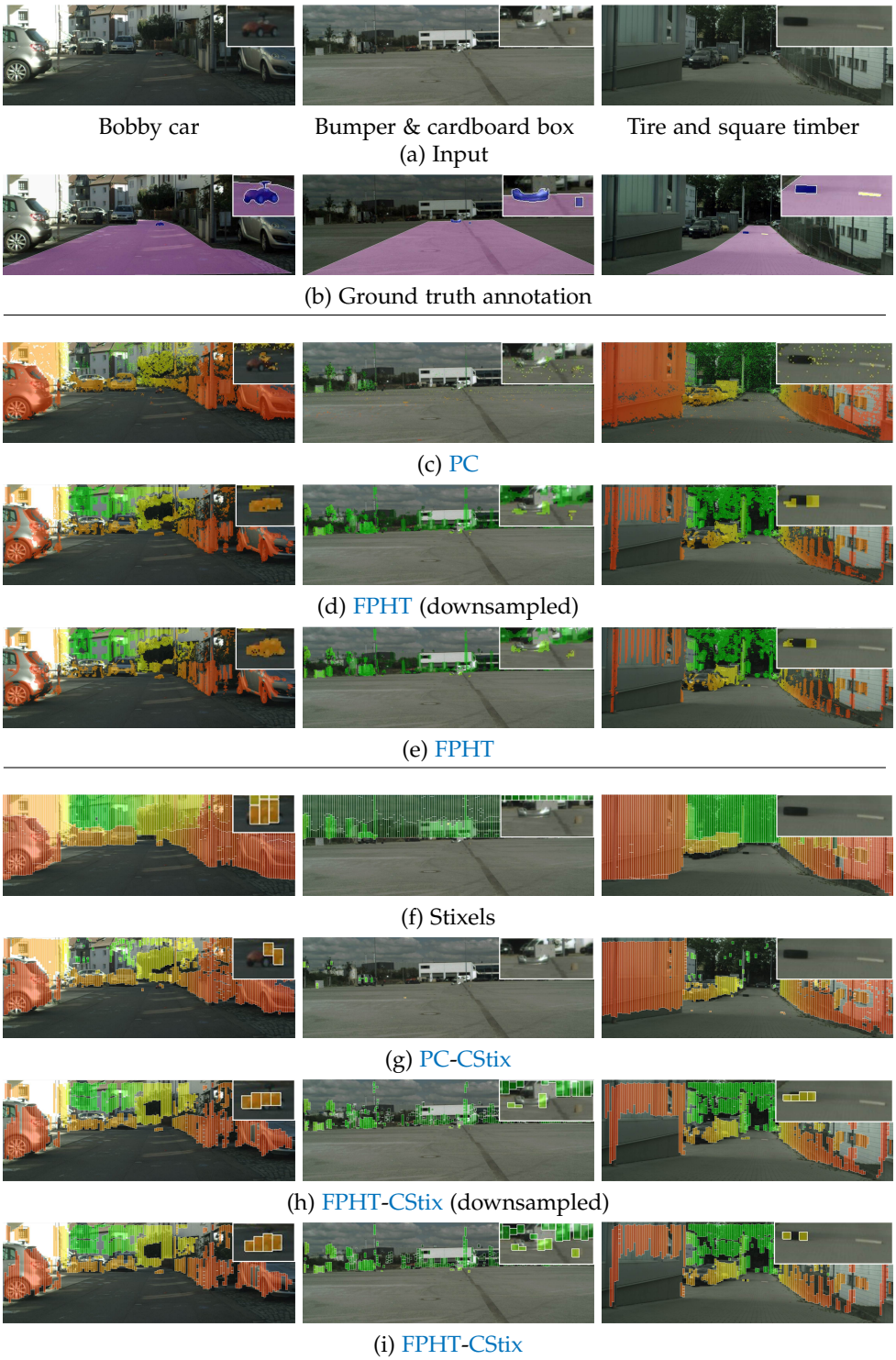


Figure 3.16: Qualitative results of the evaluated methods on the *Lost and Found* test subset. The top two rows show the left input image and the ground truth annotation, lower rows show pixel-wise and mid-level detections as overlay, color-coded by distance.

Table 3.3: Quantitative object-level results on the *Highway Detection* dataset.

	Detection Rate [%]	FP per frame	% frames with FP
Stixels [Pfeiffer and Franke, 2011]	61.4	0.79	11.5
PC-CStix [Manduchi et al., 2005]	74.5	0.56	23.0
FPHT-CStix	92.4	0.11	7.5
FPHT-BB (2D)	90.0	0.30	17.0
FPHT-BB (2D, with tracking)	87.0	0.02	2.0

the FPHT results, which illustrates the benefits of this compact representation. The second obstacle in the scene, a square timber, is not detected by any of the methods due to its low profile.

Overall, the observed qualitative results confirm that FPHT and FPHT-CStix show the best performance for various obstacles and scenarios. The PC approach suffers from increased false positives rates, since noisy disparity measurements directly influence the results. This effect could possibly be reduced by applying sophisticated spatial and temporal disparity filtering methods. The qualitative results also confirm the Stixel method's strong dependency on a correctly estimated road profile.

3.5.5.2 Highway Detection *Dataset*

In the second part of the evaluation, we analyze the object-level performance of the proposed and baseline methods on the *Highway Detection* dataset, which features a very different set of characteristics compared to the *Lost and Found* dataset. In addition to the mid-level CStix representation, we also consider a 2D Bounding Box (BB) object representation to complement FPHT. Further, given the structured traffic and regular motion models in highway scenarios, we also include a FPHT-BB variant with object tracking as described in Sect. 3.4.3.

Except for parameters depending on the dataset-specific camera characteristics, such as spatial resolution and focal length, we leave the algorithm parameters of Stixels and FPHT unchanged and reuse the working points selected in the previous section. In contrast, for PC-CStix we manually adjust the parameters to improve performance on this new dataset.

QUANTITATIVE RESULTS: OBJECT-LEVEL Again we consider the overall detection rates and false positives, as well as detection performance as a function of object distance (see Fig. 3.17).

As can be seen in Table 3.3, Stixels now produce a much more reasonable amount of false positives due to the more regular road surface. The overall detection rate lies just above 60% and is mainly limited by the long detection ranges required in this scenario. The PC-CStix approach

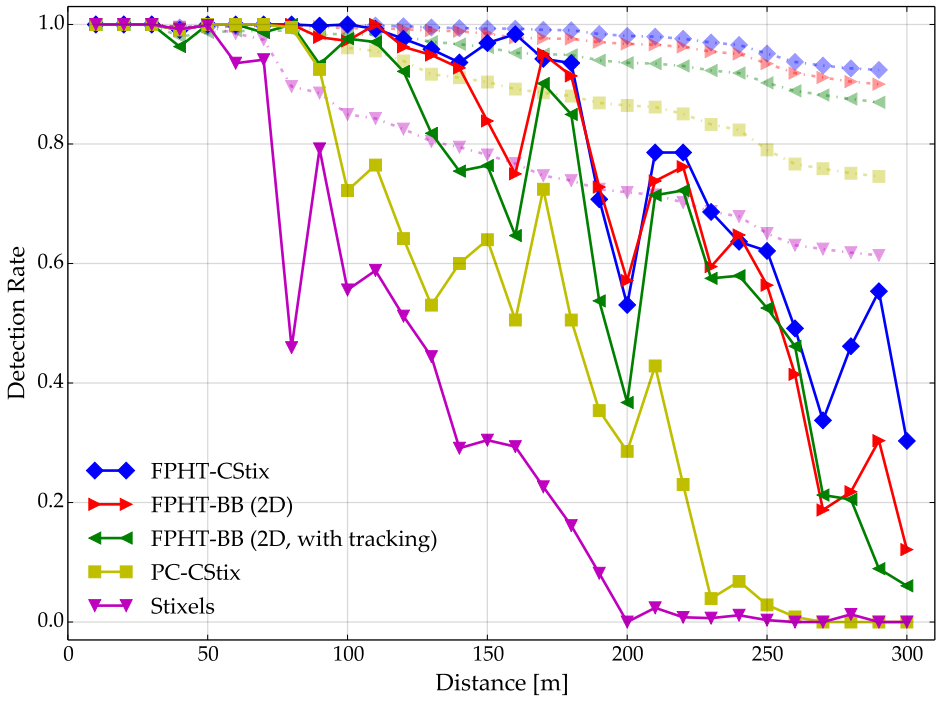
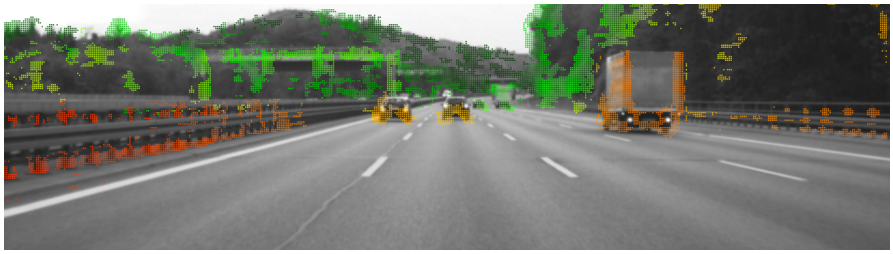


Figure 3.17: Detection rates over object distance (*Highway Detection* dataset). Solid curves illustrate the detection rates at individual range bins, dashed curves represent the integrated rate up to a given distance.

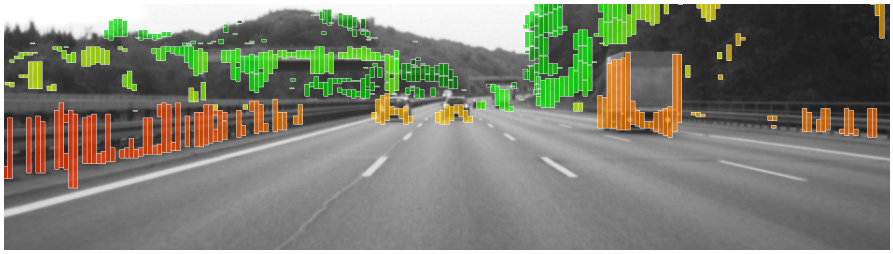
achieves a detection rate of approximately 75%, due to objects being detected at farther distances than with the Stixel algorithm. FPHT-CStix obtains the highest overall detection rate of 92.4%, with detection rates above 90% up to a distance of 180 m. The FPHT-BB variant obtains a performance slightly below FPHT-CStix, with a small decrease in detection rate and a small rise in false positives caused by sporadic errors in the bounding box fit. Here the CStix representation benefits from its flexibility, as it is not required to perform a high-quality fit of a single box to each object instance.

Integrating the FPHT-BB approach into a tracking framework allows to significantly reduce the number of false positives, since only stable object detections are passed on to the output and sporadic false detections are suppressed. As a consequence, the detection rate is also reduced slightly.

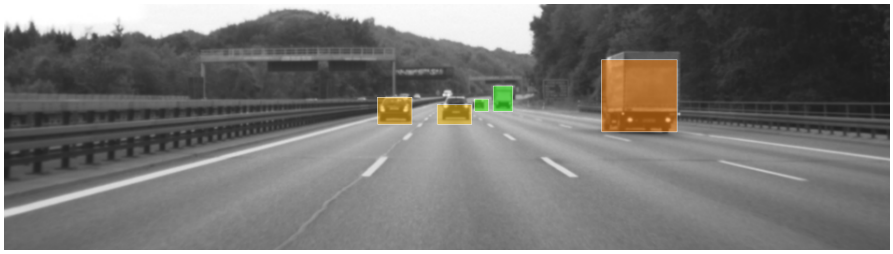
QUALITATIVE RESULTS Fig. 3.18 shows some qualitative results from the *Highway Detection* dataset. It can be seen that objects are successfully detected by FPHT at a very long range, and the results are well represented by both the CStix and the BB outputs. The bird's eye views illustrate the significant amount of noise that is present in the stereo-based long range distance estimates, even in the optimized point output of FPHT. However, the aggregation performed by CStix and BB results in a much more robust and compact estimate of object locations. This will be analyzed in more detail in Chapter 4.



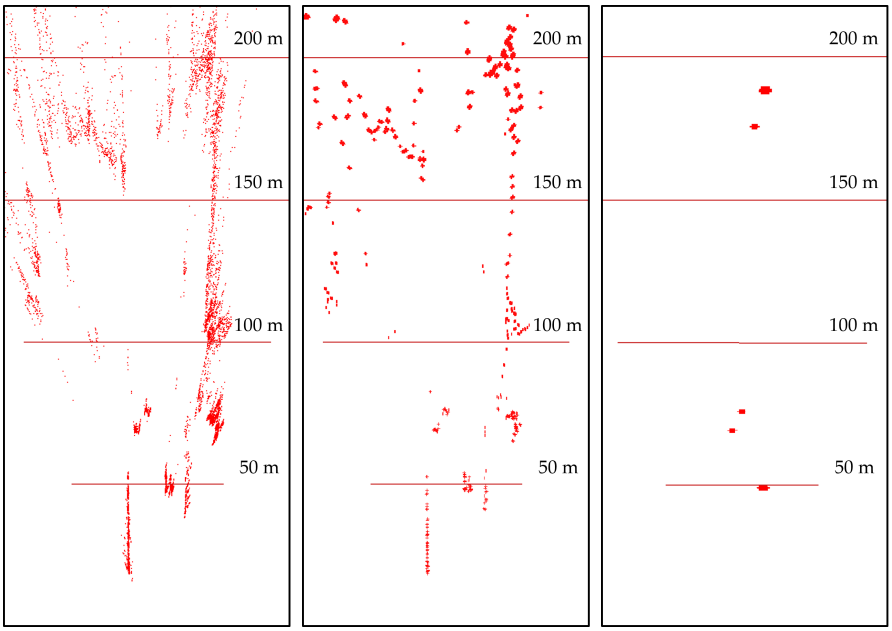
(a) FPHT



(b) FPHT-CStix



(c) FPHT-BB (2D)



(d) FPHT (e) FPHT-CStix (f) FPHT-BB (2D)

Figure 3.18: Example of objects detected by the FPHT method at distances between 50 and 200 m. (d-f) show the corresponding bird's eye views, illustrating the robust aggregation of FPHT distance estimates performed by the CStix and BB representations. Note that for the BB case only the road area ahead was considered.

3.5.5.3 *Limitations and Failure Cases*

Despite the convincing performance demonstrated in all conducted experiments, the presented PHT/FPHT detection methods naturally do have certain limitations.

First, no detections are provided in homogeneous image areas. Since the hypothesis tests are based on isolated local patches and are computed on stereo image data directly, reliable decisions are limited to textured image areas. More specifically, the approach requires sufficiently strong gradient components along the epipolar lines, i. e. horizontal intensity gradients for the used stereo camera setup. Hence, false negatives are mostly due to missing texture or insufficient object height, see e. g. Fig. 3.20. False positives sometimes occur in areas of insufficient horizontal intensity gradient, where patches barely pass the conditioning filter of Sect. 3.2.4.5. An example is shown in Fig. 3.19a. Such effects occur predominantly in areas of mixed vertical and horizontal intensity gradients along the outer image parts, probably also due to a decline in image quality and sharpness.

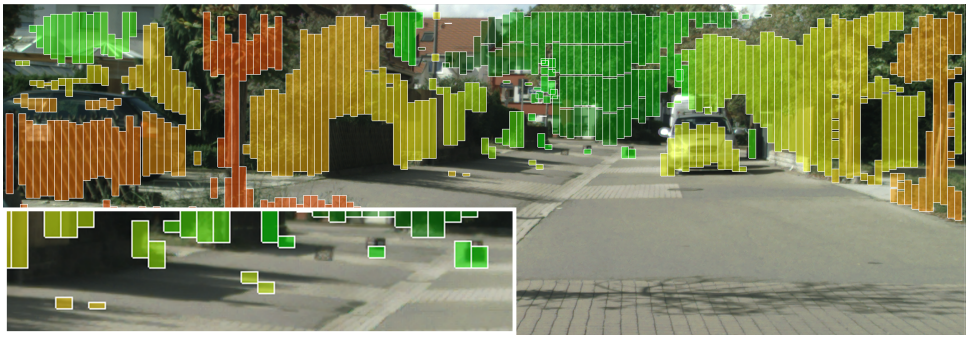
Furthermore, due to a minimum required patch size, a certain amount of foreground fattening artifacts cannot be avoided, especially in areas of homogeneous background. Here, not even the consistency checks as described in Sect. 3.2.5 can eliminate such effects. A typical example of foreground fattening artifacts can be seen on the leading vehicles in Fig. 3.18a. However, for the considered application scenarios, the critical task is the detection of an obstacle per se, whereas the estimation of the exact object dimensions is a subsequent, secondary task.

Sometimes, objects which are actually harmless to the vehicle, such as leaves or tufts of grass, might be classified as obstacles as well, see e. g. Fig. 3.19b.

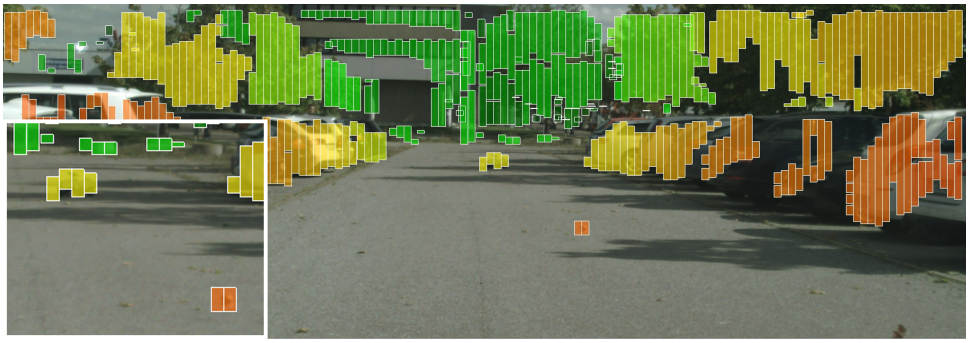
3.5.5.4 *Runtime Analysis*

Finally, we consider the runtime requirements of the PHT and FPHT algorithms. As stated previously, the design of the PHT/FPHT approaches directly allows for a significant processing speedup by straightforward parallelization. We illustrate this fact by analyzing execution times using a single CPU core, multiple CPU cores, and a GPU. Parallelization on the CPU is performed by means of OpenMP, while the full algorithm is reimplemented within the CUDA framework for execution on the GPU. The used hardware is an Intel Core i7-5960X and a Nvidia GeForce GTX Titan X, respectively.

Table 3.4 illustrates the average algorithm runtimes observed on the *Highway Detection* dataset. First of all, it can be seen that FPHT provides a speedup of four to five over PHT, independent of any applied parallelization. While the naive execution of the algorithms on a single CPU core takes several seconds per image, adding additional CPU cores yields an almost ideal speedup. Transferring the algorithms to a GPU provides a further speedup of one order of magnitude, resulting in an average

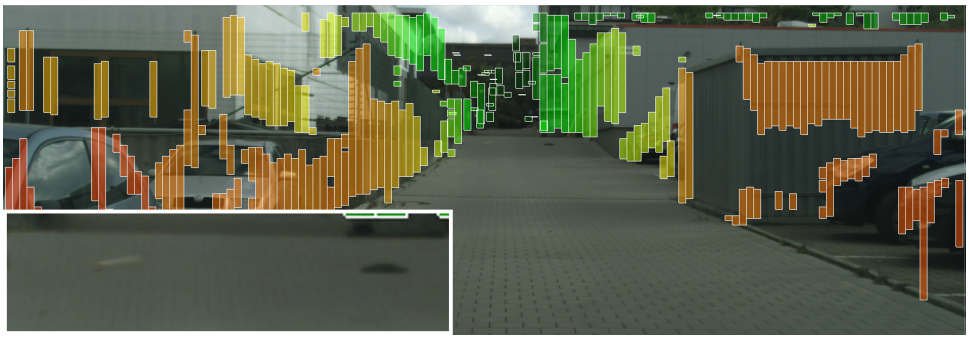


(a)

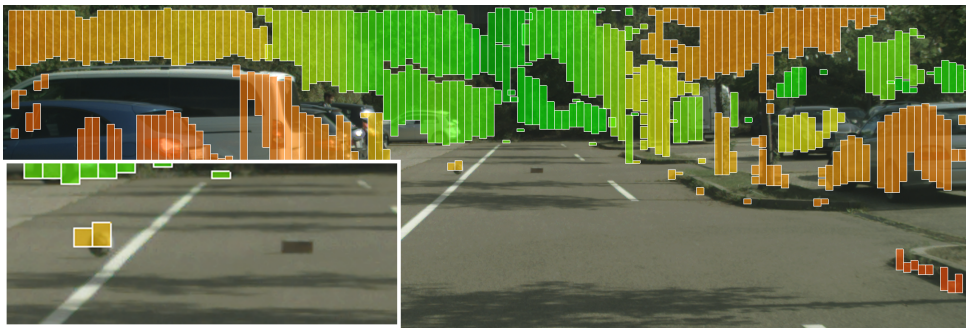


(b)

Figure 3.19: Examples of false positives generated by FPHT-CStix.



(a)



(b)

Figure 3.20: Examples of false negatives generated by FPHT-CStix.

Table 3.4: Average processing times (algorithm core) given in seconds.

	CPU (1 core)	CPU (8 cores)	GPU
PHT	12	1.5	0.12
FPHT	2.5	0.4	0.03

runtime of approximately 30 ms for the FPHT algorithm on a 1024×440 pixel image.

Note that a further speedup by code optimization or even implementation on dedicated low-power hardware such as an FPGA is conceivable.

3.6 SUMMARY

This chapter presented novel methods for stereo-based high-sensitivity generic object detection for intelligent vehicles. In particular, this includes the detection of distant or very small obstacles, without oversimplified assumptions on globally valid road geometry models. The proposed PHT and FPHT detection algorithms are based on statistical hypothesis tests using constrained, locally planar geometric hypothesis models. To achieve maximum performance, the hypothesis tests and the implicit model optimization are performed directly on image data instead of precomputed disparity maps. The algorithm core lends itself to massive parallelization, enabling real-time execution on dedicated hardware. Moreover, the concept provides a straightforward extension to multi-view camera configurations, offering the potential for a further boost in detection performance.

The PHT and FPHT approaches were tested and compared to a set of established baselines, with a focus on two critical scenarios:

- The detection of small, generic obstacles in complex urban environments.
- The detection of generic objects at long range, e. g. on highways.

In all tests on the considered challenging datasets, both PHT and FPHT significantly outperformed the selected baselines, providing a considerable increase in detection range while reducing the false positive rate at the same time. Furthermore, the proposed mid-level Cluster-Stixels (CStix) and Bounding Box (BB) object representations were proven to be very suitable for cluttered urban driving scenarios as well as more structured highway traffic.

In practice, the presented obstacle detection approaches should be applied in combination with established general-purpose 3D scene representation approaches such as the *Stixel World*. Thus, a holistic scene rep-

resentation is obtained, providing the vehicle with a comprehensive understanding of its environment while being able to handle the considered challenging cases.

To mitigate potential limitations and improve detection performance even further, a promising direction of future work is the combination of the presented high-performance geometric detection approaches with appearance-based state-of-the-art machine learning methods as recently shown in [Ramos et al., 2017].

4 DISTANCE ESTIMATION

CONTENTS

Parts of this chapter have appeared previously in [Pinggera et al., 2013] and [Pinggera et al., 2014].

4.1 INTRODUCTION

Part of the practicability and performance of modern stereo vision algorithms can arguably be attributed to the seminal Middlebury benchmark study of Scharstein and Szeliski [2002], which first provided a comprehensive framework for evaluation and enabled systematic algorithm analysis and comparison. Several years later, the KITTI project [Geiger et al., 2012, Menze and Geiger, 2015] presented a new realistic and more challenging benchmark with stereo imagery of urban traffic scenes, triggering a new wave of improved stereo vision algorithms. These major benchmark studies focus on dense stereo correspondence and are naturally required to provide *both* dense *and* accurate ground truth data. Algorithm performance is mainly judged by the percentage of pixels whose disparity estimates fall within a given accuracy threshold. The threshold is commonly set to several pixels (KITTI), or half pixels at best (Middlebury).

However, for safety-critical applications such as environment perception in autonomous driving, sub-pixel disparity accuracy is essential. Fur-

thermore, not all parts of the considered images may require the same level of attention. Obstacles in the path of motion are most relevant to the driving task, and their location and velocity have to be determined with maximum precision. Fig. 2.6 and Fig. 2.7 illustrate the significant impact of sub-pixel disparity errors on the respective distance estimates. Note that for a subsequent estimation of relative object velocities, these errors can have an even more serious influence. Unfortunately, this important aspect lies outside the scope of existing major stereo benchmarks, leaving open the question of the actually achievable disparity estimation accuracy where it matters most.

This chapter intends to fill this gap by providing an extensive statistical evaluation of object stereo matching algorithms and establishing a reference for the achievable sub-pixel accuracy limits in practice. We employ a large real-world dataset and consider various state-of-the-art stereo matching algorithms, including local differential matching and segmentation-based approaches as well as global optimization in both discrete and continuous settings. Moreover, we investigate the impact of fundamental algorithm components such as derivative filter kernels and intensity interpolation methods. Finally, we provide practical guidelines on which algorithmic aspects are essential to achieving the accuracy limits and which are not, also taking into account the trade-off between precision and computational complexity.

4.2 RELATED WORK

In major dense stereo correspondence benchmarks such as the Middlebury [Scharstein and Szeliski, 2002], Middlebury 2014 [Scharstein et al., 2014], KITTI 2012 [Geiger et al., 2012] and KITTI 2015 [Menze and Geiger, 2015] benchmarks, the number of images is kept relatively small for practical reasons, and algorithm performance is derived from pixel-wise match evaluation, weighting each pixel equally. To determine the percentage of erroneous matches, the KITTI 2012 and 2015 benchmarks employ a minimum threshold of two and three pixels, respectively. Alternatively, the average disparity error on the complete dataset can be considered, where the top-ranking algorithms at the time of writing achieve a value of 0.6 pixels [Kendall et al., 2017]. This value however provides no information on the matching accuracy for isolated salient objects.

Many top-performing dense methods make use of generic smoothness constraints on the disparity solution, either by global optimization in discrete or continuous disparity space or by integrated image segmentation and parametric model refinement. Taking a closer look at sub-pixel matching precision, it becomes clear that techniques in a discrete setting entail inherent difficulties. Sub-pixel results are obtained by fractional sampling of the disparity space and/or a curve fit to the computed matching cost volume [Szeliski and Scharstein, 2004]. Depending

on the used matching cost measure, these methods usually suffer from the so-called pixel-locking effect, i.e. an uneven sub-pixel disparity distribution. Various approaches have been proposed to alleviate this effect, including two-stage shifted matching [Shimizu and Okutomi, 2001], symmetic refinement [Nehab et al., 2005], design of optimal cost interpolation functions [Haller and Nedevschi, 2012] and disparity smoothing filters [Gehrig and Franke, 2007]. In contrast, methods set in a continuous framework [Ranftl et al., 2012] or based on segment model fitting [Vogel et al., 2015] do not suffer from pixel-locking and have been shown to outperform discrete techniques with regard to sub-pixel accuracy.

When shifting the focus from dense disparity maps to isolated objects, the properties of local area-based matching techniques have to be investigated. Within the context of image registration, Robinson and Milanfar [2004] presented a comprehensive analysis of the fundamental accuracy limits under simple translatory motion. In low noise conditions, iterative differential matching methods based on [Lucas and Kanade, 1981] were shown to reach errors of below 1/100 pixels. The corresponding Cramer-Rao Lower Bound (CRLB) for registration errors turns out to be a combination of noise and bias terms, with bias being caused by suboptimal methods for image derivative estimation and image interpolation as well as mathematical approximations. Similar results were reported in [Sutton et al., 2009] for stereoscopic high-precision strain analysis applications. The optimal design of derivative filters and interpolation kernels was also identified as an essential issue in optical flow [Scharr, 2007], super-resolution [Elad et al., 2005], and medical imaging [Farid and Simoncelli, 2004, Thévenaz et al., 2000] literature.

Perhaps most relevant to the present work is a recent study on local stereo block matching accuracy by Sabater et al. [2011]. In contrast to the work mentioned above, realistic noise conditions were investigated and a theoretical formulation for the expected disparity error was derived. Results from a phase correlation local matching algorithm were shown to agree with the presented theory, demonstrating an accuracy of down to 1/20 pixel on pre-selected pixel locations. However, experiments were performed only on a set of three synthetic stereo pairs and the four classic Middlebury images.

An important aspect, but outside the scope of the present object-based statistical evaluation, is the data-driven pre-selection of reliable matching points. For local differential methods, matching accuracy can be predicted based on the local image structure [Förstner, 1993]. Point selection methods based on various confidence measures have been explored for local [Sabater et al., 2012] as well as global methods [Pfeiffer et al., 2013].

4.3 METHODS

All algorithms considered here assume a calibrated stereo camera setup and rectified image pairs. For each relevant object in the scene, a single representative disparity value is determined. This makes sense in the

considered scenario, where it is sufficient to model the visible relevant objects as fronto-parallel planes. Note that at large distances, where accurate disparity estimation is actually most important, this model is also valid for more general scenarios.

For the purpose of this study, the object detections, i.e. approximate image location and size, are given in advance. Corresponding rectangular patches in the left stereo images are provided as input to the matching algorithms (see Fig. 4.6). Details on the generation of these object patches are described in Sect. 4.4.3.

4.3.1 *Local Differential Matching (LDM)*

Iterative local differential matching methods, originally proposed by Lucas and Kanade [1981] and Tomasi and Kanade [1991], have proven to perform exceptionally well at high-accuracy displacement estimation of image patches [Robinson and Milanfar, 2004, Sutton et al., 2009, Pinggera et al., 2013]. Notably, the PHT and FPHT object detection approaches presented in Sect. 3 share the same underlying direct differential matching concept. However, for the present task the warp $\vec{W}$ is reduced to a simple shift of the horizontal image coordinates, representing the constant stereo disparity d of all pixels $\vec{x}$ inside an image patch Ω:

$$\vec{W} = \vec{W}(\vec{x}, d) = \begin{pmatrix} x - d \\ y \end{pmatrix}. \tag{4.1}$$

As in Sect. 3.2.3 we formulate a statistical image formation model, considering the discrete left and right image patch values $I_l(\vec{x})$ and $I_r(\vec{x})$ as noisy samples of the observed continuous image intensity signal f at position $\vec{x}$. The terms $\alpha_l(\vec{x})$ and $\alpha_r(\vec{x})$ model a potential local intensity offset, while $\eta_l(\vec{x})$ and $\eta_r(\vec{x})$ represent noise samples from a zero-mean normal distribution with an assumed variance σ^2:

$$I_l(x, y) = f(x, y) + \alpha_l(x, y) + \eta_l(x, y) \tag{4.2}$$

$$I_r(x - d, y) = f(x, y) + \alpha_r(x, y) + \eta_r(x, y). \tag{4.3}$$

First, to compensate for a potential local offset, the mean intensity for each considered patch is removed. Treating the intensity values of all pixels in the patch area Ω as observations of f with additive i.i.d. noise, from (4.3) we obtain an expression for the negative log-likelihood of a certain disparity d:

$$\ln\left(p(\vec{\mathcal{I}}; d)\right) = \sum_{\vec{x} \in \Omega} C_1 - C_2 \cdot \left(I_r(x - d, y) - f(x, y)\right)^2, \tag{4.4}$$

where C_1 and C_2 are constants. To obtain the optimal disparity estimate $\hat{d}$ for the patch, we aim to find the Maximum Likelihood Estimate (MLE), which corresponds to minimizing the cost function F:

$$F(d) = \sum_{\vec{x} \in \Omega} r^2(\vec{x}, d) = \sum_{\vec{x} \in \Omega} \left(I_r(x - d, y) - f(x, y) \right)^2 , \tag{4.5}$$

$$\hat{d} = \arg\min_{d} \left(F(d) \right) . \tag{4.6}$$

As described in Sect. 3.2.4.4, in practice (4.2) is first solved for f and plugged into (4.3), effectively replacing the unknown image signal f in the cost function by the reference image samples I_l. The non-linear optimization problem (4.6) is then solved via appropriate iterative algorithms such as Gauss-Newton (see Sect. 2.1).

4.3.1.1 *Inverse Compositional Formulation*

Similar to the derivation in Sect. 3.3.3, the LDM approach can be reformulated to exploit the inverse compositional algorithm proposed by Baker and Matthews [2004]. By reversing the roles of the input images and introducing compositional parameter updates, the computational load of solving (4.6) is reduced. Even more importantly, as shown by Sutton et al. [2009], matching bias which occurs with the original LDM formulation is reduced by the inverse compositional approach. The required signal derivatives are estimated only once at integer pixel positions and do not have to be warped in each iteration. In this way, errors resulting from interpolating derivative kernel responses are avoided. Consequently, we use the inverse compositional formulation as the default LDM implementation.

The matching problem (4.6) is cast into the compositional formulation, where the warp of (4.1) is updated incrementally by $\vec{W} \leftarrow \vec{\odot} \left(\vec{W}, \vec{W}(\vec{x}, \Delta d) \right)$, with

$$\vec{\odot} \left(\vec{W}, \vec{W}(\vec{x}, \Delta d) \right) \equiv \vec{W} \left(\vec{W}(\vec{x}, \Delta d), d \right) = \begin{pmatrix} x - \Delta d - d \\ y \end{pmatrix} . \tag{4.7}$$

Using the inverse compositional approach to switch the roles of the input images, we can rewrite the cost to be minimized at each step as

$$F \left(\vec{W}, \vec{W}(\vec{x}, \Delta d) \right) = \sum_{\vec{x} \in \Omega} \left(I_r(\vec{W}) - f(\vec{W}(\vec{x}, \Delta d)) \right)^2 . \tag{4.8}$$

The inverse warp update is then performed according to $\vec{W} \leftarrow \vec{\odot}\left(\vec{W}, \vec{W}(\vec{x}, \Delta d)^{-1}\right)$, where

$$\vec{\odot}\left(\vec{W}, \vec{W}(\vec{x}, \Delta d)^{-1}\right) = \begin{pmatrix} x + \Delta d - d \\ y \end{pmatrix}. \tag{4.9}$$

The gradient and hence the Jacobian of the cost function with respect to the disparity update is then

$$\begin{aligned} \vec{g}^T\left(\vec{W}, \vec{W}(\vec{x}, \Delta d)\right) &= \mathbf{J}_{F \circ \vec{W}}\left(\vec{W}, \vec{W}(\vec{x}, \Delta d)\right) \\ &= \sum_{\vec{x} \in \Omega} 2\left(I_r(\vec{W}) - f(\vec{W}(\vec{x}, \Delta d))\right) \\ &\cdot \frac{\partial\left(I_r(\vec{W}) - f(\vec{W}(\vec{x}, \Delta d))\right)}{\partial\left(W_x(\vec{x}, \Delta d), W_y(\vec{x}, \Delta d)\right)} \\ &\cdot \frac{d\left(W_x(\vec{x}, \Delta d), W_y(\vec{x}, \Delta d)\right)}{d\Delta d} \end{aligned} \tag{4.10}$$

with

$$\begin{aligned} \frac{\partial\left(I_r(\vec{W}) - f(\vec{W}(\vec{x}, \Delta d))\right)}{\partial\left(W_x(\vec{x}, \Delta d), W_y(\vec{x}, \Delta d)\right)} &= -\begin{pmatrix} \frac{\partial f\left(\vec{W}(\vec{x}, \Delta d)\right)}{\partial W_x(\vec{x}, \Delta d)} \\ \frac{\partial f\left(\vec{W}(\vec{x}, \Delta d)\right)}{\partial W_y(\vec{x}, \Delta d)} \end{pmatrix}^T \\ &= -\vec{\nabla} f^T\left(\vec{W}(\vec{x}, \Delta d)\right) \end{aligned} \tag{4.11}$$

and

$$\frac{d\left(W_x(\vec{x}, \Delta d), W_y(\vec{x}, \Delta d)\right)}{d\Delta d} = \begin{pmatrix} \frac{dW_x(\vec{x}, \Delta d)}{d\Delta d} \\ \frac{dW_y(\vec{x}, \Delta d)}{d\Delta d} \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}. \tag{4.12}$$

The cost function gradient at $\Delta d = 0$ is thus

$$\begin{aligned} \vec{g}^T\left(\vec{W}\right) &= \vec{g}^T\left(\vec{W}, \vec{W}(\vec{x}, \Delta d)\right)\Big|_{\Delta d = 0} \\ &= \sum_{\vec{x} \in \Omega} 2\left(I_r(\vec{W}) - f(\vec{x})\right) \cdot \nabla f_x(\vec{x}) \\ &= 2 \sum_{\vec{x} \in \Omega} r\left(\vec{W}, \vec{x}\right) \cdot \nabla f_x(\vec{x}). \end{aligned} \tag{4.13}$$

Note that the gradient of f has to be evaluated at the original pixel position $\vec{x}$ only. Finally, the approximate Gauss-Newton Hessian at $\Delta d = 0$ is

$$\mathbf{A}_{GN}\left(\vec{W}\right) = 2 \sum_{\vec{x} \in \Omega} \left(\nabla f_x(\vec{x})\right)^2, \tag{4.14}$$

Table 4.1: Separable pre-smoothing and derivative filter kernels. Complement symmetric and antisymmetric values respectively.

	Pre-smoothing filter	Derivative filter
Scharr 3×3	$[\ldots, 0.5450, 0.2275]$	$[\ldots, 0, 0.5]$
Scharr 5×5	$[\ldots, 0.4260, 0.2493, 0.0377]$	$[\ldots, 0, 0.2767, 0.1117]$
Central Diff. 5×5	$[\ldots, 0.40260.2442, 0.0545]$	$\frac{1}{12}[1, -8, 0, 8, -1]$

which is independent of the current disparity estimate and hence can be precomputed and reused in each iteration. The incremental disparity update Δd is obtain by solving

$$-\vec{g}^T\left(\vec{W}\right) = \mathbf{A}_{GN}\left(\vec{W}\right)\Delta d. \tag{4.15}$$

With this we have the inverse compositional formulation of the classical local differential matching algorithm of Kanade, Lucas and Tomasi.

4.3.1.2 *Image Derivative Estimation*

In practice, the exact signal derivatives required for solving (4.15) are not known and have to be approximated from the image data using discrete derivative filters. However, inexact derivatives lead to matching bias as shown by Robinson and Milanfar [2004] and Elad et al. [2005]. To minimize such errors, the use of optimal filter kernels is necessary. Jähne [1995] derived an optimized second order central differences kernel which requires a separate smoothing step for signal bandwidth limitation. Farid and Simoncelli [2004] and Scharr [2007] on the other hand proposed the joint optimization of pairs of signal pre-smoothing and derivative filters. We investigate both methods, using 3×3 and 5×5 Scharr kernels as well as a 5×5 central difference kernel with a $5 \times 5/\sigma = 1$ Gaussian pre-smoother, see Table 4.1.

4.3.1.3 *Image Interpolation*

Even when using the inverse compositional formulation, the iterative nature of the LDM approach still requires warping the right image patch I_r in each iteration. Naturally, this step involves the evaluation of intensity values at sub-pixel positions and therefore makes a suitable image interpolation method necessary. In previous studies on image interpolation, for example by Thévenaz et al. [2000], approaches based on B-Spline representations clearly outperformed simpler methods such as cubic convolution [Keys, 1981] and bilinear interpolation. We investigate the impact of interpolation on disparity accuracy, with cubic B-Splines following [Unser et al., 1993] as the reference method.

4.3.1.4 *Symmetric Matching (LDM+)*

Following the considerations of Sect. 3.2.4.4, it can be seen that in the above formulation of LDM the observations I_l and I_r of the image signal f are not treated symmetrically. Consequently, we introduce a symmetric LDM+ algorithm to evaluate its impact on the accuracy of the resulting disparity estimate. To this end, an estimate $\hat{f}$ of the unknown image signal is computed in conjunction with the optimal disparity $\hat{d}$. The disparity is computed according to Sect. 4.3.1.1, while $\hat{f}$ is re-estimated in each iteration as the mean of the respectively aligned input image pixels. Note that in contrast to Sect. 3.2.4.4, now due to the inverse compositional formulation the required signal derivatives are effectively computed from both input images by applying the derivative filters of Sect. 4.3.1.2 directly to the current estimate of f.

In the case of LDM+, the computational advantages gained from the inverse compositional formulation cannot be exploited since here the respective terms again have to be recomputed in each iteration.

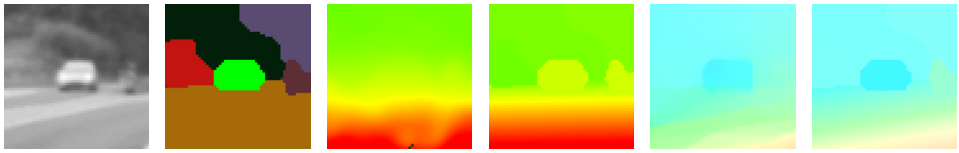
4.3.2 *Joint Matching and Segmentation (MSEG)*

Common local matching techniques, such as the LDM algorithm, inherently make the assumption that all pixels in the input image patches conform to a single simple displacement model. Outliers corresponding to a different model can significantly distort estimation results. To overcome this problem we propose a Joint Matching and Segmentation (MSEG) algorithm, previously presented in [Pinggera et al., 2013]. The approach reduces errors due to outliers by jointly optimizing both the patch shape and the corresponding parametric displacement model. A probabilistic multi-cue formulation integrating disparity, optical flow and pixel intensity distributions is proposed to reliably segment the relevant object from its surroundings. At the same time the iterative approach refines disparity and optical flow parameters in a LDM manner.

4.3.2.1 *Probabilistic Multi-Cue Segmentation*

Our goal is to find a precise binary segmentation of a given image patch, separating the relevant object from the background, and at the same time to perform an accurate estimation of the disparity of the object. However, in the considered scenarios the background can in general not be described accurately by a single disparity, optical flow or intensity model. Instead, a representation using several separate segments is required. An example can be seen in Fig. 4.1, where road plane, background vegetation and road infrastructure have to be distinguished for a correct representation of the scene.

Each segment $k \in K$ is described by its pixel support Ω_k, parametric models for disparity d_k and optical flow $\vec{v}_k = \left(v_{kx},\ v_{ky}\right)^T$, as well as a non-parametric intensity model i_k.



(a) Input (b) MSEG segments (c) SGM disparity (d) MSEG disparity (e) TV-L1 flow (f) MSEG flow

Figure 4.1: Example of the MSEG algorithm applied to a car observed at approx. 150 m distance (a). The proposed multi-cue approach yields a correct segmentation of object and background (b) as well as an accurate local disparity map (d) and optical flow field (f) compared to sub-pixel smoothed SGM of Gehrig et al. [2012] (c) and TV-L1 flow of Wedel et al. [2009b] (e).

We consider the different segmentation cues as realizations of conditionally independent random fields with the respective probability densities $p(\vec{v}|\vec{\mathcal{I}},\ell)$, $p(d|\vec{\mathcal{I}},\ell)$ and $p(i|\vec{\mathcal{I}},\ell)$, dependent on the segmentation or labeling ℓ and the input image data $\vec{\mathcal{I}}$[1]. While the assumption of independence between disparity and optical flow might not hold in general scenes, we assume it to be approximately fulfilled in our case of very small disparity ranges, where flow magnitudes are dominated by rotational camera motion and independent object motion. The posterior probability distribution of all possible labelings can then be described in a Bayesian manner:

$$\begin{aligned} p(\ell|\vec{\mathcal{I}},\vec{v},d,i) &= \frac{p(\vec{\mathcal{I}},\vec{v},d,i|\ell)\cdot p(\ell)}{p(\vec{\mathcal{I}},\vec{v},d,i)} \\ &= \frac{p(\vec{v},d,i|\vec{\mathcal{I}},\ell)\cdot p(\ell|\vec{\mathcal{I}})\cdot p(\vec{\mathcal{I}})}{p(\vec{v},d,i|\vec{\mathcal{I}})\cdot p(\vec{\mathcal{I}})} \\ &\approx \frac{p(\vec{v}|\vec{\mathcal{I}},\ell)\cdot p(d|\vec{\mathcal{I}},\ell)\cdot p(i|\vec{\mathcal{I}},\ell)\cdot p(\ell|\vec{\mathcal{I}})}{p(\vec{v},d,i|\vec{\mathcal{I}})}. \end{aligned} \tag{4.16}$$

The sought-for segmentation corresponds to the Maximum a Posteriori (MAP) estimate of ℓ. We employ an Expectation-Maximization (EM) scheme to iteratively refine both the segmentation and the respective segment parameters.

DISPARITY AND OPTICAL FLOW AS RANDOM FIELDS To formulate optical flow $\vec{v}$ and disparity d directly as random fields with parametric probability distributions, we draw on the approach of Cremers and Yuille [2003] and Schoenemann and Cremers [2006], originally designed for motion segmentation only. The deviations of the apparent flow $\tilde{\vec{v}}_k(\vec{x})$ and disparity $\tilde{d}_k(\vec{x})$ at pixel $\vec{x}$ from the corresponding model-based values $\vec{v}_k(\vec{x})$ and $d_k(\vec{x})$ of segment k are considered as realizations η of in-

1 Note that $\vec{\mathcal{I}}$ includes the stereo image pair at the current time step t as well as the image pair of the previous time step $t-1$.

dependent and normally distributed random variables with zero mean and variance $\vec{\sigma}^2_{\vec{v}_k} = ({\sigma_{\vec{v}_k}}^2_x, {\sigma_{\vec{v}_k}}^2_y)^T$ and $\sigma^2_{d_k}$, respectively:

$$\tilde{\vec{v}}_k(\vec{x}) - \vec{v}_k(\vec{x}) = \vec{\eta}_{\vec{v}}(\vec{x}) = \left(\eta_{\vec{v}x}(\vec{x}), \quad \eta_{\vec{v}y}(\vec{x})\right)^T \tag{4.17}$$

$$\tilde{d}_k(\vec{x}) - d_k(\vec{x}) = \eta_d(\vec{x}), \tag{4.18}$$

For simplicity we assume ${\sigma_{\vec{v}_k}}^2_x = {\sigma_{\vec{v}_k}}^2_y = \sigma^2_{\vec{v}_k}$.[2]

We assume that by applying the apparent displacement values $\tilde{\vec{v}}_k(\vec{x})$ and $\tilde{d}_k(\vec{x})$, the corresponding image residuals r vanish:

$$r_{\vec{v}}(\vec{x}, \tilde{\vec{v}}_k) = I_l(\vec{x}) - I_l^{t-1}(\vec{x} + \tilde{\vec{v}}_k(\vec{x})) \stackrel{!}{=} 0 \tag{4.19}$$

$$r_d(\vec{x}, \tilde{d}_k) = I_l(\vec{x}) - I_r(x - \tilde{d}_k(\vec{x}), y) \stackrel{!}{=} 0, \tag{4.20}$$

where I_l^{t-1} denotes the left stereo image of the previous time step.

Given the availability of reasonable initial estimates $\vec{v}_k^-$ and d_k^-, flow and disparity can be written as compositions of these initial estimates and corresponding differential updates:

$$\vec{v}_k(\vec{x}) = \vec{v}_k^-(\vec{x}) + \Delta\vec{v}_k(\vec{x}), \qquad \tilde{\vec{v}}_k(\vec{x}) = \vec{v}_k^-(\vec{x}) + \Delta\tilde{\vec{v}}_k(\vec{x}) \tag{4.21}$$

$$d_k(\vec{x}) = d_k^-(\vec{x}) + \Delta d_k(\vec{x}), \qquad \tilde{d}_k(\vec{x}) = d_k^-(\vec{x}) + \Delta\tilde{d}_k(\vec{x}). \tag{4.22}$$

The formulation of (4.17) and (4.18) can then be rewritten as

$$\Delta\tilde{\vec{v}}_k(\vec{x}) - \Delta\vec{v}_k(\vec{x}) = \vec{\eta}_{\vec{v}}(\vec{x}) \tag{4.23}$$

$$\Delta\tilde{d}_k(\vec{x}) - \Delta d_k(\vec{x}) = \eta_d(\vec{x}). \tag{4.24}$$

Approximating the image residuals by a first-order Taylor series yields

$$\begin{aligned} r_{\vec{v}}(\vec{x}, \tilde{\vec{v}}_k) &\approx I_l(\vec{x}) - I_l^{t-1}(\vec{x} + \vec{v}_k^-) - \vec{\nabla} I_l^{t-1}(\vec{x} + \vec{v}_k^-) \cdot \Delta\tilde{\vec{v}}_k(\vec{x}) \\ &= r_{\vec{v}}(\vec{x}, \vec{v}_k^-) - \vec{\nabla} I_l^{t-1}(\vec{x} + \vec{v}_k^-) \cdot (\Delta\vec{v}_k(\vec{x}) + \vec{\eta}_{\vec{v}}(\vec{x})) \\ &\stackrel{!}{=} 0 \end{aligned} \tag{4.25}$$

and

$$\begin{aligned} r_d(\vec{x}, \tilde{d}_k) &\approx I_l(\vec{x}) - I_r(\vec{x} - d_k^-) + \nabla I_{rx}(\vec{x} - d_k^-) \cdot \Delta\tilde{d}_k(\vec{x}) \\ &= r_d(\vec{x}, d_k^-) + \nabla I_{rx}(\vec{x} - d_k^-) \cdot (\Delta d_k(\vec{x}) + \eta_d(\vec{x})) \\ &\stackrel{!}{=} 0. \end{aligned} \tag{4.26}$$

2 An alternative is to apply the noise model to the length of the flow vector only and hence use the image gradient in the direction of the flow vector in the subsequent approximation of the residuals.

Finally, through (4.25) and (4.26) we can derive expressions for computing the likelihoods of $\vec{v}_k$ and d_k at pixel $\vec{x}$, given the input data and the image segmentation ℓ:

$$p(\vec{v}_k(\vec{x})|\vec{\mathcal{I}},\ell) = \frac{1}{\sqrt{2\pi\sigma^2_{\vec{v}_k}}} \cdot \exp\left(-\frac{\left(\vec{\nabla} I_l^{t-1\,T}(\vec{x}+\vec{v}_k^-)\Delta\vec{v}_k(\vec{x}) - r_{\vec{v}}(\vec{x},\vec{v}_k^-)\right)^2}{2\cdot|\vec{\nabla} I_l^{t-1}(\vec{x}+\vec{v}_k^-)|^2\cdot\sigma^2_{\vec{v}_k}}\right) \tag{4.27}$$

$$p(d_k(\vec{x})|\vec{\mathcal{I}},\ell) = \frac{1}{\sqrt{2\pi\sigma^2_{d_k}}} \cdot \exp\left(-\frac{\left(\nabla I_{rx}(\vec{x}-d_k^-)\Delta d_k(\vec{x}) + r_d(\vec{x},d_k^-)\right)^2}{2\cdot|\nabla I_{rx}(\vec{x}-d_k^-)|^2\cdot\sigma^2_{d_k}}\right). \tag{4.28}$$

The corresponding variances $\sigma^2_{\vec{v}_k}$ and $\sigma^2_{d_k}$ are computed as

$$\sigma^2_{\vec{v}_k} = \frac{1}{|\Omega_k|}\cdot\sum_{\vec{x}\in\Omega_k}\frac{\left(\vec{\nabla} I_l^{t-1\,T}(\vec{x}+\vec{v}_k^-)\Delta\vec{v}_k(\vec{x}) - r_{\vec{v}}(\vec{x},\vec{v}_k^-)\right)^2}{|\vec{\nabla} I_l^{t-1}(\vec{x}+\vec{v}_k^-)|^2} \tag{4.29}$$

$$\sigma^2_{d_k} = \frac{1}{|\Omega_k|}\cdot\sum_{\vec{x}\in\Omega_k}\frac{\left(\nabla I_{rx}(\vec{x}-d_k^-)\Delta d_k(\vec{x}) + r_d(\vec{x},d_k^-)\right)^2}{|\nabla I_{rx}(\vec{x}-d_k^-)|^2}. \tag{4.30}$$

INTENSITY DISTRIBUTION The intensity model of each segment is described by a non-parametric probability distribution $p(i_k|\vec{\mathcal{I}},\ell)$, which allows to represent general cases with multiple modes. A kernel density estimation is used to approximate the intensity distributions from the pixel values within the segment support regions Ω_k.

SEGMENTATION Computing the MAP estimate labeling by solving $\hat{\ell} = \arg\max_\ell\left(p(\ell|\vec{\mathcal{I}},\vec{v},d,i)\right)$ is equivalent to minimizing the respective negative log-likelihood, i.e. the negative logarithm of the posterior $p(\ell|\vec{\mathcal{I}},\vec{v},d,i)$ as defined in (4.16):

$$\hat{\ell} = \arg\min_\ell\left(-\log\left(p(\ell|\vec{\mathcal{I}},\vec{v},d,i)\right)\right). \tag{4.31}$$

In order to employ a useful model for the prior term $p(\ell|\vec{\mathcal{I}})$ and at the same time keep the problem tractable, we assume a pixel-wise first-order Markov property and represent $p(\ell|\vec{\mathcal{I}},\vec{v},d,i)$ in the form of an undirected factor graph, more precisely a Conditional Random Field (CRF). The set of sites in the graph corresponds to the pixels of the image patch Ω, with the set of neighbors of a pixel p denoted as $\mathcal{N}_p$ and the set of

cliques of size n as $\mathcal{C}_n$. We define $\mathcal{N}_p$ to describe a common spatial eight-neighborhood with a maximal clique size of $n = 2$. The assigned scene element label at p is denoted as l_p, where $l_p \in K$.

According to the Hammersley-Clifford theorem [Hammersley and Clifford, 1971, Besag, 1974], in order for the joint posterior $p(\ell|\vec{\mathcal{I}}, \vec{v}, d, i)$ to satisfy the assumed Markov properties, it has to take the form of a Gibbs distribution

$$p(\ell|\vec{\mathcal{I}}, \vec{v}, d, i) = \frac{1}{Z} \exp\left(-E(\ell|\vec{\mathcal{I}}, \vec{v}, d, i)\right), \tag{4.32}$$

where Z represents the partition function, a normalizing constant, and E denotes the energy function defined by the clique potentials of the graph. Taking the negative logarithm of (4.32) yields

$$-\log\left(p(\ell|\vec{\mathcal{I}}, \vec{v}, d, i)\right) = E(\ell|\vec{\mathcal{I}}, \vec{v}, d, i) + \log(Z), \tag{4.33}$$

which shows that solving (4.31) to determine $\hat{\ell}$ is equivalent to minimizing the energy E. Note that Z and the denominator in $p(\ell|\vec{\mathcal{I}}, \vec{v}, d, i)$ have no influence on the location of the minimum and can be ignored in the MAP estimation task.

The energy to be minimized can be expressed as a sum of clique potentials V over all possible cliques [Li, 2010]:

$$\begin{aligned} E(\ell|\vec{\mathcal{I}}, \vec{v}, d, i) &= \sum_{c \in \mathcal{C}} V_c(\ell|\vec{\mathcal{I}}, \vec{v}, d, i) \\ &= \sum_{p \in \mathcal{C}_1} V_1(l_p|\vec{\mathcal{I}}, \vec{v}, d, i) + \sum_{p,q \in \mathcal{C}_2} V_2(l_p, l_q|\vec{\mathcal{I}}) \\ &= \sum_{p \in \Omega} V_1(l_p|\vec{\mathcal{I}}, \vec{v}, d, i) + \sum_{p \in \Omega} \sum_{q \in \mathcal{N}_p} V_2(l_p, l_q|\vec{\mathcal{I}}) \\ &= E_{data}(\ell|\vec{\mathcal{I}}, \vec{v}, d, i) + E_{prior}(\ell|\vec{\mathcal{I}}). \end{aligned} \tag{4.34}$$

The data term E_{data} consists of unary potentials only and is fully defined by the pixel-wise observation likelihoods derived previously:

$$\begin{aligned} V_1(l_p|\vec{\mathcal{I}}, \vec{v}, d, i) = &-\log\left(p(\vec{v}_{l_p}(\vec{x}_p))|\vec{\mathcal{I}}, \ell)\right) \\ &-\log\left(p(d_{l_p}(\vec{x}_p)|\vec{\mathcal{I}}, \ell)\right) \\ &-\log\left(p(i_{l_p}(\vec{x}_p)|\vec{\mathcal{I}}, \ell)\right). \end{aligned} \tag{4.35}$$

The term E_{prior} corresponding to the label prior $p(\ell|\vec{\mathcal{I}})$ comprises the binary potentials, which are formulated as

$$V_2(l_p, l_q|\vec{\mathcal{I}}) = \begin{cases} \gamma \cdot \left(\beta + (1-\beta)\exp\left(\frac{|I(\vec{x}_p)-I(\vec{x}_q)|}{\sigma}\right)\right) \cdot \frac{1}{|\vec{x}_p-\vec{x}_q|} & l_p \neq l_q \\ 0 & l_p = l_q \end{cases}. \quad (4.36)$$

We apply a common contrast-sensitive cost function as in [Boykov and Funka-Lea, 2006] to encourage smoothness in homogeneous regions and label discontinuities at high image gradients. The parameter γ is used to balance data and prior terms.

Within the data term E_{data}, the variances of the disparity and flow model of each segment serve as implicit inverse weighting factors. Considering (4.29) and (4.30), the displacement estimates of homogeneous image segments will in general tend to have higher variances and therefore less influence on the location of the energy minimum. Conversely, such segments will show more discriminative peaks in their intensity distributions, and vice versa.

To efficiently compute a high quality approximate solution to the energy minimization problem, we use the alpha-expansion graph cut approach of [Boykov and Kolmogorov, 2004, Boykov et al., 2001].

PARAMETER UPDATE Given the result of the segmentation step, the parameters of each segment are updated. To parametrize flow and disparity, either a translational or an affine parameter model is assigned to each segment. We use affine parameter models to approximate slanted surfaces in the world which cannot be reduced to fronto-parallel planes even at large distances.

The optical flow vector at each pixel is hence defined as $\vec{v}_k(\vec{x}) = \mathbf{C}_{\vec{v}}(\vec{x})\vec{\vartheta}_k$ and the disparity as $d_k(\vec{x}) = \mathbf{C}_d(\vec{x})\vec{\delta}_k$, with

$$\mathbf{C}_{\vec{v},transl} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; \mathbf{C}_{\vec{v},affine}(\vec{x}) = \begin{pmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{pmatrix}, \quad (4.37)$$

$$\mathbf{C}_{d,transl} = 1, \qquad \mathbf{C}_{d,affine}(\vec{x}) = \begin{pmatrix} x, & y, & 1 \end{pmatrix} \quad (4.38)$$

and

$$\vec{\vartheta}_{k,transl} = \begin{pmatrix} v_{kx}, & v_{ky} \end{pmatrix}^T, \quad \vec{\vartheta}_{k,affine} = \begin{pmatrix} \vec{\vartheta}_k^{(1)}, & \dots, & \vec{\vartheta}_k^{(6)} \end{pmatrix}^T, \quad (4.39)$$

$$\vec{\delta}_{k,transl} = d, \qquad \vec{\delta}_{k,affine} = \begin{pmatrix} \vec{\delta}_k^{(1)}, & \dots, & \vec{\delta}_k^{(3)} \end{pmatrix}^T. \quad (4.40)$$

The parameter update is then computed by setting the respective partial derivatives of the energy formulation (4.34) to zero and solving the resulting normal equations $\mathbf{A}_{\vec{v}} \Delta\vec{\vartheta}_k = \vec{b}_{\vec{v}}$ and $\mathbf{A}_d \Delta\vec{\delta}_k = \vec{b}_d$ for $\Delta\vec{\vartheta}_k$ and $\Delta\vec{\delta}_k$, respectively, where

$$\mathbf{A}_{\vec{v}} = \sum_{\Omega_k} \frac{\mathbf{C}_{\vec{v}}^T \vec{\nabla} I_l^{t-1} \vec{\nabla} I_l^{t-1\,T} \mathbf{C}_{\vec{v}}}{|\vec{\nabla} I_l^{t-1}|^2}, \tag{4.41}$$

$$\vec{b}_{\vec{v}} = -\sum_{\Omega_k} \frac{\mathbf{C}_{\vec{v}}^T \vec{\nabla} I_l^{t-1} \cdot r_{\vec{v}}(\vec{v}_k^-)}{|\vec{\nabla} I_l^{t-1}|^2}, \tag{4.42}$$

and

$$\mathbf{A}_d = \sum_{\Omega_k} \mathbf{C}_d^T \mathbf{C}_d, \tag{4.43}$$

$$\vec{b}_d = -\sum_{\Omega_k} \frac{\mathbf{C}_d^T \nabla I_{rx} \cdot r_d(d_k^-)}{|\nabla I_{rx}|^2}. \tag{4.44}$$

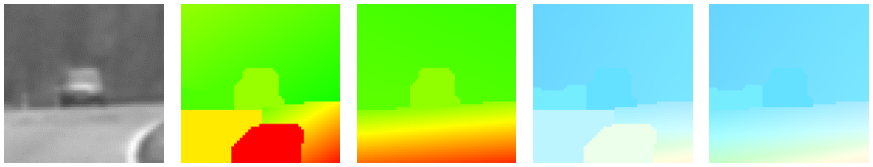
Here the location arguments for evaluating the image gradient and the residuals have been omitted for brevity.

The update step is repeated as long as the energy decreases and the length of the update vector lies above a given threshold. The respective images are iteratively warped with the current parameter estimates and the additive updates $\vec{\vartheta}_k^- \leftarrow \vec{\vartheta}_k^- + \Delta\vec{\vartheta}_k$ and $\vec{\delta}_k^- \leftarrow \vec{\delta}_k^- + \Delta\vec{\delta}_k$ are computed as described above.

Finally, the variances of the parameter models of each segment are estimated according to (4.29) and (4.30).

The updates of the non-parametric intensity models $p(i_k|\vec{\mathcal{I}}, \ell)$ are simply computed from the intensity values of the pixel support of each segment.

GLOBAL PRIORS Computing local displacement parameter estimates can yield erroneous results in homogeneous image areas. Fig. 4.2 shows an example where the segment parameter solution diverges in the featureless regions of the road plane. Here global correspondence algorithms benefit from strong regularization, propagating values from more reliable image areas outside the local image patch. Hence, we describe how prior flow and disparity results, provided by dense global algorithms such as TV-L1 [Wedel et al., 2009b] or SGM [Gehrig et al., 2015], can be included into our approach to constrain the local parameter solution. All results derived in the following for optical flow can be simply transferred to the case of one-dimensional displacement for the disparity parameters.



(a) Input (b) No prior (c) SGM prior (d) No prior (e) TV prior

Figure 4.2: Integrating priors from global algorithms prevents diverging local disparity and flow estimates in featureless regions such as the road plane.

To integrate the prior flow results $\vec{v}_p$ into our probabilistic model, $p(\vec{v}|\vec{\mathcal{I}}, \ell)$ in (4.16) is replaced by $p(\vec{v}|\vec{v}_p, \vec{\mathcal{I}}, \ell)$:

$$p(\vec{v}|\vec{v}_p, \vec{\mathcal{I}}, \ell) = \frac{p(\vec{v}_p|\vec{v}, \vec{\mathcal{I}}, \ell) \cdot p(\vec{v}|\vec{\mathcal{I}}, \ell)}{p(\vec{v}_p|\vec{\mathcal{I}}, \ell)}. \tag{4.45}$$

The term $p(\vec{v}_p|\vec{v}, \vec{\mathcal{I}}, \ell)$ represents the likelihood of the prior flow $\vec{v}_p$ being in accordance with the locally computed values $\vec{v}$. Since the prior flow is independent of the segmentation, the constant factor $p(\vec{v}_p|\vec{\mathcal{I}}, \ell)$ can be dropped in the optimization. Formulating $p(\vec{v}_p|\vec{v}, \vec{\mathcal{I}}, \ell)$ as a normal distribution with mean $\vec{v}$ independently for each pixel allows to use its variance $\vec{\sigma}^2_{\vec{v}_p}(\vec{x}) = \begin{pmatrix} {\sigma_{\vec{v}_p}}_x^2(\vec{x}) & {\sigma_{\vec{v}_p}}_y^2(\vec{x}) \end{pmatrix}^T$ to locally define the desired influence of the prior, depending on the image gradient magnitude $|\vec{\nabla} I_l|$ at each pixel, normalized by its local mean:

$${\sigma_{\vec{v}_p}}_x^2(\vec{x}) = \alpha_{prior} \cdot \left(\frac{|\nabla I_{l_x}|}{mean\left(|\nabla I_{l_x}|\right)} + 1 \right)^2, \tag{4.46}$$

$${\sigma_{\vec{v}_p}}_y^2(\vec{x}) = \alpha_{prior} \cdot \left(\frac{|\nabla I_{l_y}|}{mean\left(|\nabla I_{l_y}|\right)} + 1 \right)^2. \tag{4.47}$$

This rather heuristic choice is motivated by the desire that in homogeneous regions with low image gradients the confidence in the global prior should be higher than in the locally computed parameters while at object edges and in structured regions the local solution should be allowed to deviate from the prior.

Solving for the flow parameters as before yields

$$\mathbf{A}_{\vec{v}} = \sum_{\Omega_k} \left(\frac{\mathbf{C}^T_{\vec{v}} \vec{\nabla} I_l^{t-1} \vec{\nabla} I_l^{t-1\,T} \mathbf{C}_{\vec{v}}}{|\vec{\nabla} I_l^{t-1}|^2 \cdot \sigma^2_{\vec{v}_k}} + \frac{\mathbf{C}^T_{\vec{v}} \vec{f}_x \vec{f}_x{}^T \mathbf{C}_{\vec{v}}}{{\sigma_{\vec{v}_p}}_x^2} + \frac{\mathbf{C}^T_{\vec{v}} \vec{f}_y \vec{f}_y{}^T \mathbf{C}_{\vec{v}}}{{\sigma_{\vec{v}_p}}_y^2} \right), \tag{4.48}$$

$$\vec{b}_{\vec{v}} = -\sum_{\Omega_k} \left(\frac{\mathbf{C}_{\vec{v}}^T \vec{\nabla} I_l^{t-1} \cdot r_{\vec{v}}(\vec{v}_k^-)}{|\vec{\nabla} I_l^{t-1}|^2 \cdot \sigma_{\vec{v}_k}^2} + \frac{\mathbf{C}_{\vec{v}}^T \vec{f}_x \cdot (v_{p_x} - v_{k_x}^-)}{\sigma_{\vec{v}_{p_x}}^2} + \frac{\mathbf{C}_{\vec{v}}^T \vec{f}_y \cdot (v_{p_y} - v_{k_y}^-)}{\sigma_{\vec{v}_{p_y}}^2} \right), \tag{4.49}$$

with $\vec{f}_x = \left(\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right)$, $\vec{f}_y = \left(\begin{smallmatrix} 0 \\ -1 \end{smallmatrix}\right)$, and flow results $\vec{v}_k^-$ of the previous iteration. This represents a least squares parameter solution with priors similar to the one in [Baker et al., 2004] but with additional adaptive weighting factors depending on both $\sigma_{\vec{v}_k}^2$ and the selected $\vec{\sigma}_{\vec{v}_p}^2(\vec{x})$.

PARAMETER REFINEMENT After the segmentation algorithm has converged, the LDM+ approach as described above is applied in an additional parameter refinement step to obtain the final disparity estimate $\hat{d}$. The result thus combines the accuracy of the LDM+ approach with the optimized support regions and initial values obtained from the MSEG algorithm.

4.3.2.2 *Scene Flow Segmentation (MSEG+)*

In order to investigate the impact of exploiting the full data from two consecutive stereo pairs, we extend the MSEG approach by introducing an additional scene flow segmentation constraint and using all four images for disparity refinement as presented in [Pinggera et al., 2014].

In the original MSEG algorithm derived above, a Gaussian noise model is applied directly to the disparity $d_k(\vec{x})$ and optical flow vectors $\vec{v}_k(\vec{x})$ at each pixel, which allows for the formulation of probabilistic segmentation criteria by regarding the apparent values $\tilde{d}_k(\vec{x})$ and $\tilde{\vec{v}}_k(\vec{x})$ as realizations of conditionally independent random variables given $\vec{\mathcal{I}}, \ell$.

Now a scene flow constraint is added to couple the disparity displacements d between left and right stereo images at the current time step t with the optical flow vectors $\vec{v}$ between consecutive left images [Rabe, 2011], while the respective degradations due to noise are still considered to be conditionally independent. The constraint complementing (4.19) and (4.20) is expressed as

$$r_{\vec{s}}(\vec{x}, \tilde{\vec{v}}_k, \tilde{d}_k) = I_l^{t-1}(\vec{x} + \tilde{\vec{v}}_k(\vec{x})) - I_r(x - \tilde{d}_k(\vec{x}), y) \stackrel{!}{=} 0. \tag{4.50}$$

Approximation by a first-order Taylor series yields

$$\begin{aligned}
&r_{\vec{s}}(\vec{x}, \tilde{\vec{v}}_k, \tilde{d}_k) \\
&\quad \approx I_l^{t-1}(\vec{x} + \vec{v}_k^-) + \vec{\nabla} I_l^{t-1}(\vec{x} + \vec{v}_k^-) \cdot \Delta\tilde{\vec{v}}_k(\vec{x}) \\
&\qquad - I_r(\vec{x} - d_k^-) + \nabla I_{rx}(\vec{x} - d_k^-) \cdot \Delta\tilde{d}_k(\vec{x}) \\
&\quad = r_{\vec{s}}(\vec{x}, \vec{v}_k^-, d_k^-) \\
&\qquad + \vec{\nabla} I_l^{t-1}(\vec{x} + \vec{v}_k^-) \cdot (\Delta\vec{v}_k(\vec{x}) + \vec{\eta}_{\vec{v}}(\vec{x})) \\
&\qquad + \nabla I_{rx}(\vec{x} - d_k^-) \cdot (\Delta d_k(\vec{x}) + \eta_d(\vec{x})) \\
&\quad = r_{\vec{s}}(\vec{x}, \vec{v}_k^-, d_k^-) \\
&\qquad + \vec{\nabla} I_l^{t-1}(\vec{x} + \vec{v}_k^-)\Delta\vec{v}_k(\vec{x}) \\
&\qquad + \nabla I_{rx}(\vec{x} - d_k^-)\Delta d_k(\vec{x}) + \eta_{\vec{s}}(\vec{x}) \\
&\quad \stackrel{!}{=} 0,
\end{aligned} \tag{4.51}$$

where the noise term $\eta_{\vec{s}}(\vec{x})$ with variance $\sigma_{\vec{s}_k}^2 = |\nabla I_{rx}(\vec{x} - d_k^-)|^2 \cdot \sigma_{d_k}^2 + \vec{\nabla} I_l^{t-1}(\vec{x} + \vec{v}_k^-)^2 \cdot \sigma_{\vec{v}_k}^2$ stems from the assumed degradation models of $\tilde{\vec{v}}_k$ and $\tilde{d}_k$. Based on (4.51) the likelihood of the scene flow $\vec{s}$ given $\vec{v}_k$ and d_k at each pixel can be computed as

$$\begin{aligned}
p(\vec{s}_k(\vec{x}) | \vec{v}_k, d_k, \vec{\mathcal{I}}, \ell) &= \frac{1}{\sqrt{2\pi\sigma_{\vec{s}_k}^2}} \cdot \\
\exp\left(- \frac{\left(\vec{\nabla} I_l^{t-1\,T} \Delta\vec{v}_k(\vec{x}) + \nabla I_{rx} \Delta d_k(\vec{x}) + r_{\vec{s}}(\vec{x}, \vec{v}_k^-, d_k^-) \right)^2}{2 \cdot \sigma_{\vec{s}_k}^2} \right),
\end{aligned} \tag{4.52}$$

where $\vec{\nabla} I_l^{t-1}$ and ∇I_{rx} are evaluated at $\vec{x} + \vec{v}_k^-$ and $\vec{x} - d_k^-$, respectively. The optimized patch shape is then determined by assigning optimal segment models for pixel intensity i, disparity d and optical flow $\vec{v}$ under the scene flow constraint, thus maximizing the segmentation likelihood

$$\begin{aligned}
&p(\ell | \vec{\mathcal{I}}, \vec{s}, \vec{v}, d, i) \\
&\quad \approx \frac{p(\vec{s} | \vec{v}, d, \vec{\mathcal{I}}, \ell) \cdot p(\vec{v} | \vec{\mathcal{I}}, \ell) \cdot p(d | \vec{\mathcal{I}}, \ell) \cdot p(i | \vec{\mathcal{I}}, \ell) \cdot p(\ell | \vec{\mathcal{I}})}{p(\vec{s}, \vec{v}, d, i | \vec{\mathcal{I}})}.
\end{aligned} \tag{4.53}$$

For the joint update, the parameter vectors $\vec{\vartheta}_k$ and $\vec{\delta}_k$ used in MSEG are now combined into $\vec{\theta}_k$ such that

$$\begin{pmatrix} v_{kx}(\vec{x}) \\ v_{ky}(\vec{x}) \\ d_k(\vec{x}) \end{pmatrix} = \mathbf{C}(\vec{x})\vec{\theta}_k, \tag{4.54}$$

with

$$\mathbf{C}_{transl} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{4.55}$$

$$\mathbf{C}_{affine}(\vec{x}) = \begin{pmatrix} x & y & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & y & 1 \end{pmatrix}, \tag{4.56}$$

and

$$\vec{\theta}_{k,transl} = \begin{pmatrix} v_{kx}, & v_{ky}, & d_k \end{pmatrix}^T, \tag{4.57}$$

$$\vec{\theta}_{k,affine} = \begin{pmatrix} \vartheta_{k1}, & \dots, & \vartheta_{k6}, & \delta_{k1}, & \dots, & \delta_{k3} \end{pmatrix}^T. \tag{4.58}$$

The parameter update is then again obtained by setting the respective partial derivatives of the overall energy to zero and solving the resulting normal equations $\mathbf{A}\Delta\vec{\theta}_k = \vec{b}$ for $\Delta\vec{\theta}_k$, where

$$\mathbf{A} = \sum_{\Omega_k} \mathbf{C}^T \mathbf{M} \mathbf{C}, \tag{4.59}$$

$$\vec{b} = -\sum_{\Omega_k} \mathbf{C}^T \vec{h}, \tag{4.60}$$

with

$$\mathbf{M}(\vec{x}) = \begin{pmatrix} \frac{1}{\sigma_{d_k}^2} + \frac{{\nabla I_{rx}}^2}{\sigma_{\vec{s}_k}^2} & \frac{\nabla I_{rx} \nabla I_{l_x}^{t-1}}{\sigma_{\vec{s}_k}^2} & \frac{\nabla I_{rx} \nabla I_{l_y}^{t-1}}{\sigma_{\vec{s}_k}^2} \\ \frac{\nabla I_{rx} \nabla I_{l_x}^{t-1}}{\sigma_{\vec{s}_k}^2} & \frac{{\nabla I_{l_x}^{t-1}}^2}{|\vec{\nabla} I_l^{t-1}|^2 \sigma_{\vec{v}_k}^2} + \frac{{\nabla I_{l_x}^{t-1}}^2}{\sigma_{\vec{s}_k}^2} & \frac{\nabla I_{l_x}^{t-1} \nabla I_{l_y}^{t-1}}{|\vec{\nabla} I_l^{t-1}|^2 \sigma_{\vec{v}_k}^2} + \frac{\nabla I_{l_x}^{t-1} \nabla I_{l_y}^{t-1}}{\sigma_{\vec{s}_k}^2} \\ \frac{\nabla I_{rx} \nabla I_{l_y}^{t-1}}{\sigma_{\vec{s}_k}^2} & \frac{\nabla I_{l_x}^{t-1} \nabla I_{l_y}^{t-1}}{|\vec{\nabla} I_l^{t-1}|^2 \sigma_{\vec{v}_k}^2} + \frac{\nabla I_{l_x}^{t-1} \nabla I_{l_y}^{t-1}}{\sigma_{\vec{s}_k}^2} & \frac{{\nabla I_{l_y}^{t-1}}^2}{|\vec{\nabla} I_l^{t-1}|^2 \sigma_{\vec{v}_k}^2} + \frac{{\nabla I_{l_y}^{t-1}}^2}{\sigma_{\vec{s}_k}^2} \end{pmatrix} \tag{4.61}$$

and

$$\vec{h}(\vec{x}) = \begin{pmatrix} \frac{\nabla I_{rx} \cdot r_d(d_k^-)}{|\nabla I_{rx}|^2 \sigma_{d_k}^2} + \frac{\nabla I_{rx} \cdot r_{\vec{s}}(\vec{v}_k^-, d_k^-)}{\sigma_{\vec{s}_k}^2} \\ \frac{\nabla I_{l_x}^{t-1} \cdot r_{\vec{v}}(\vec{v}_k^-)}{|\vec{\nabla} I_l^{t-1}|^2 \sigma_{\vec{v}_k}^2} + \frac{\nabla I_{l_x}^{t-1} \cdot r_{\vec{s}}(\vec{v}_k^-, d_k^-)}{\sigma_{\vec{s}_k}^2} \\ \frac{\nabla I_{l_y}^{t-1} \cdot r_{\vec{v}}(\vec{v}_k^-)}{|\vec{\nabla} I_l^{t-1}|^2 \sigma_{\vec{v}_k}^2} + \frac{\nabla I_{l_y}^{t-1} \cdot r_{\vec{s}}(\vec{v}_k^-, d_k^-)}{\sigma_{\vec{s}_k}^2} \end{pmatrix}. \tag{4.62}$$

Having obtained an optimized patch shape, for the final disparity refinement step we again resort to the LDM+ algorithm, now aligning all four input images to estimate the unknown signal f and its gradient.

4.3.3 *Multi-LDM (M-LDM)*

In the LDM and MSEG approaches described above, each relevant object is represented by exactly one image patch, and the corresponding disparity estimate is computed by solving (4.6) using all pixels of the patch. To ensure that only true object pixels are considered during optimization and no outliers distort the result, MSEG performs a dedicated segmentation task. As a relatively simple and efficient alternative, we also consider a so-called Multi-LDM or M-LDM approach, where the disparity of a single object is estimated from solving multiple LDM problems. Several *mini-patches* of size 7×7 pixels are distributed across the actual object patch and the disparity of each one is computed independently via LDM. The final object disparity is then computed by combining all patch results. We use the interquartile mean to obtain a robust estimate of the object disparity.

Additionally, we consider a variant called M-LDM 2D, which extends M-LDM by estimating vertical displacement in addition to the stereo disparity. This essentially corresponds to computing a two-dimensional optical flow vector between the two stereo images for each mini-patch. The aim of the M-LDM 2D approach is to improve invariance with regard to calibration inaccuracies resulting in vertical displacement.

4.3.4 *Fast Direct Planar Hypothesis Testing (FPHT)*

Continuing along the lines of the M-LDM approach, we go one step further and directly use the FPHT method as proposed in Sect. 3.3 to not only detect but simultaneously compute an accurate disparity estimate of all relevant objects. Since the FPHT algorithm inherently optimizes all object hypothesis positions, the final disparity of a detected object can directly be obtained as in the M-LDM approach, by taking the robust mean of all object points provided by FPHT. Considering an object-level bounding box presentation as described in Sect. 3.4.2, the disparity is computed from the inliers of the bounding box representation.

4.3.5 *Total Variation Stereo (TV)*

As a representative for global stereo matching approaches in a continuous setting, we investigate a differential matching algorithm with variational optimization. Total Variation (TV)-based algorithms, originally designed for optical flow estimation [Werlberger et al., 2009, Wedel et al., 2009b], have been shown to perform very well in stereo applications [Ranftl et al., 2012]. Specifically, we use a Total Variation Huber-L1 stereo im-

plementation of Rabe [2011] adapted from [Werlberger et al., 2009]. The algorithm uses an iterative pyramidal approach to globally optimize an energy of the form

$$E = \int \int \lambda |I_r(x-d,y) - I_l(x,y)| + \sum_{k=1}^{2} \rho\left(|\nabla d_k|\right)\ dy\ dx, \tag{4.63}$$

where the regularization term $\rho\left(|\nabla d_k|\right)$ penalizes the spatial variation of disparity values, using the robust Huber loss with threshold t_h. For algorithm details we refer to [Werlberger et al., 2009]. We set $t_h = 0.01$, $\lambda = 25$ and use five image pyramid levels. For robustness with regard to changes in illumination, the structure-texture decomposition of Wedel et al. [2009b] is applied.

While the resulting dense disparity map provided by the global algorithm is useful for many applications, an additional processing step is needed to arrive at representative disparity values for isolated objects. We compute the interquartile mean of the pixel disparities within the input image patch to obtain a robust object disparity estimate for evaluation.

4.3.5.1 *Symmetric Matching (TV+)*

Since the variational approach makes use of the same differential matching principle as the local LDM method on a pixel-wise basis, the symmetry-considerations of the LDM+ modification can also be applied. We include a TV+ variant which performs the joint estimation of both displacement and unknown image signal at each iteration. To estimate the required image derivatives, a 3×3 Scharr kernel is used.

4.3.6 *Semi-Global Matching (SGM)*

Finally, we evaluate the discrete Semi-Global Matching (SGM) algorithm of Hirschmüller [2008]. The method approximates a two-dimensional optimization with truly global constraints by first computing pixel-wise matching costs and then applying one-dimensional regularization along paths from eight directions at each pixel. The nature of the approach allows for efficient computation, and we make use of a fast implementation on specialized hardware as presented in [Gehrig et al., 2009].

While all algorithms described above perform matching using image intensities directly, here we employ the census transform and corresponding Hamming distances as a matching cost. This provides a very robust algorithm suitable for challenging real-world scenarios [Gehrig et al., 2009, 2015]. Sub-pixel results are computed by a symmetric V-fit to three adjacent values in the regularized matching cost volume [Haller and Nedevschi, 2012]. Again, we compute the interquartile mean to obtain object disparities.

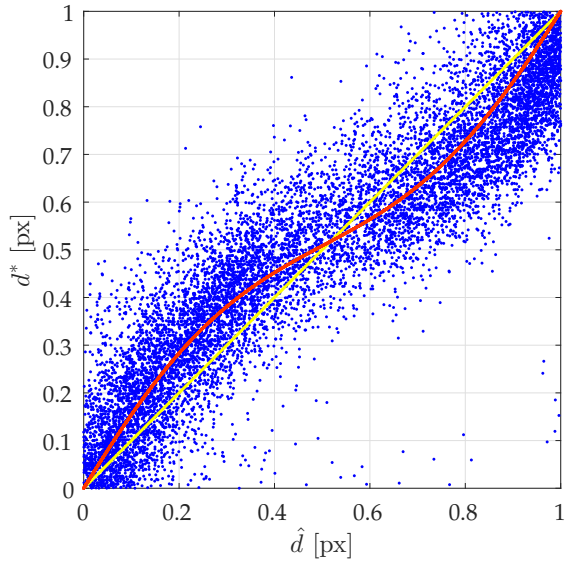


Figure 4.3: Distribution of ground truth disparities d^* over values $\hat{d}$ estimated via SGM in the sub-pixel interval $[0, 1]$. The pixel-locking effect results in a skewed measurement distribution, which we represent by a low-order polynomial (red curve). The model is used to generate a look-up table for online compensation.

4.3.6.1 *Pixel-Locking Compensation (SGM+PLC)*

As mentioned previously, matching methods in a discrete setting suffer from the so-called pixel-locking effect, i.e. a biased distribution of sub-pixel disparity values. The severity of this effect depends on the used cost metric. While the census transform provides robust matching results, the associated pixel-locking effect is particularly prominent. For general stereo applications, different methods to alleviate the effect have been presented [Shimizu and Okutomi, 2001, Haller and Nedevschi, 2012]. However, for the scenario at hand we propose a straightforward and efficient post-processing step, which largely neutralizes object-based pixel-locking errors. With ground truth data for the desired object disparities available, the systematic sub-pixel bias can be estimated from a set of raw measurements directly. To this end we project both expected and measured disparity values into the sub-pixel interval $[0, 1]$ and fit a low-order polynomial to the resulting two-dimensional point cloud. This curve is stored and directly provides the necessary offsets for an efficient online correction of the object disparities (see Fig. 4.3).

4.4 EVALUATION

This section presents an in-depth evaluation of the matching accuracy achievable by the various algorithms introduced above. First, the required performance metrics are defined, followed by a description of the dedicated dataset created for this study. Then the evaluation

methodology is described, and finally a thorough analysis of the results is presented.

4.4.1 *Evaluation Metrics*

4.4.1.1 *Disparity Error*

The disparity error ϵ_d represents the deviation of the estimated stereo disparity $\hat{d}$ from the ground truth value d^* at each frame:

$$\epsilon_d = \hat{d} - d^*. \tag{4.64}$$

4.4.1.2 *Temporal Disparity Error Variation*

The disparity error ϵ_d as described above provides an absolute accuracy measure for all object observations, combining the measurements of multiple unique objects. However, it alone does not provide sufficient information on the relative accuracy for a single tracked object over time. This is essential if the velocities of individual objects are to be determined. In this case, the relative accuracy between consecutive measurements of the object of interest is just as important as e.g. a potential constant disparity bias.

To describe the object-based relative measurement accuracy over time, independent of a potential disparity bias, we define $\nabla\epsilon_d$ as the disparity error variation using finite differences:

$$\nabla\epsilon_d = \epsilon_{d,t} - \epsilon_{d,t-1}. \tag{4.65}$$

4.4.1.3 *Robust Statistics*

The distributions of ϵ_d and $\nabla\epsilon_d$ are examined both over the complete dataset and as a function of absolute distance. To obtain robust estimates $\overline{\epsilon_d}$ and $\overline{\nabla\epsilon_d}$ of the respective mean values, we make use of the interquartile mean. Moreover, we compute robust estimates of scale to assess the statistical variability of the measurements, corresponding to the respective standard deviations. To this end, we employ the location-free scale estimator S_n of Rousseeuw and Croux [1993], which has a breakdown point of 50% and a Gaussian efficiency of 58%. S_n is computed by considering the pair-wise differences between all measured samples, indicated in the following by i and j:

$$S_n(\epsilon_d) = 1.1926 \cdot median_i \left(median_j \left(|\epsilon_{d_i} - \epsilon_{d_j}| \right) \right), \tag{4.66}$$

$$S_n(\nabla\epsilon_d) = 1.1926 \cdot median_i \left(median_j \left(|\nabla\epsilon_{d_i} - \nabla\epsilon_{d_j}| \right) \right). \tag{4.67}$$

4.4.2 *Dataset*

A central aspect of the present evaluation is the use of an extensive dataset to allow for a meaningful statistical analysis. Furthermore, we exclusively use real-world data to be able to draw conclusions most relevant for practical applications.

The dataset previously presented in [Pinggera et al., 2014] consists of 70,000 grayscale image pairs recorded from a vehicle-mounted stereo camera system in highway scenarios during mostly sunny weather conditions. The setup exhibits a baseline length of 38 centimeters and a focal length of 1240 pixels, with spatial and radiometric resolutions of 1024×440 pixels and 12 bits, respectively. Ground truth is provided by a long range RADAR sensor. Owing to its underlying measurement principle, RADAR is able to determine longitudinal distances of isolated moving objects with high precision. The used reference sensor yields a measurement uncertainty of $3\sigma \cong 0.5$ m over the full considered distance range.

In the present work, we introduce some slight modifications to the dataset used in [Pinggera et al., 2014]. Since the measurement cycles of the reference RADAR sensor and the stereo camera system are not identical, not all corresponding measurements are perfectly aligned in time. Hence, in [Pinggera et al., 2014] the reference data was interpolated to still obtain valid reference measurements for each stereo image pair. Here, we use a more stringent concept by considering only the subset of the data with a guaranteed consistent temporal offset and minimizing this offset across the whole dataset. While this method reduces the number of individual measurements available for evaluation, it guarantees the quality of the remaining synchronized data. Fig. 4.4 illustrates the implications of deviating from the optimized temporal offset value, i. e. the impact on the apparent matching accuracy.

The updated dataset features approximately 200 unique vehicles representing relevant objects, yielding a total of more than 12,000 disparity measurements for evaluation. The distribution of object observations over absolute distance in the dataset is visualized in Fig. 4.5.

4.4.3 *Methodology*

To detect relevant objects in the images and provide them as fair input to the stereo algorithm evaluation, we apply a combined vehicle detection and tracking method as described in [Franke et al., 2013]. A strong mixture-of-experts classifier inspired by [Enzweiler and Gavrila, 2011] is used to detect vehicles at the required large distances. The vehicles are then tracked over time, accumulating confidence in the process. For evaluation we consider objects which have been tracked for more than 15 frames. The objects are represented by a rectangular patch in the left stereo image, two examples can be seen in Fig. 4.6.

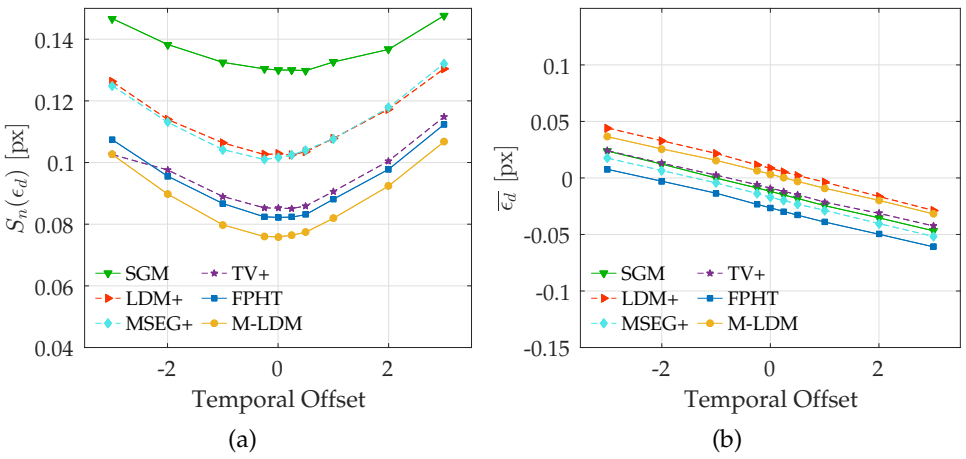


Figure 4.4: Optimized synchronization (offset = 0) between ground truth (RADAR) and stereo measurements. Temporal offsets are given in units of image acquisition cycles.

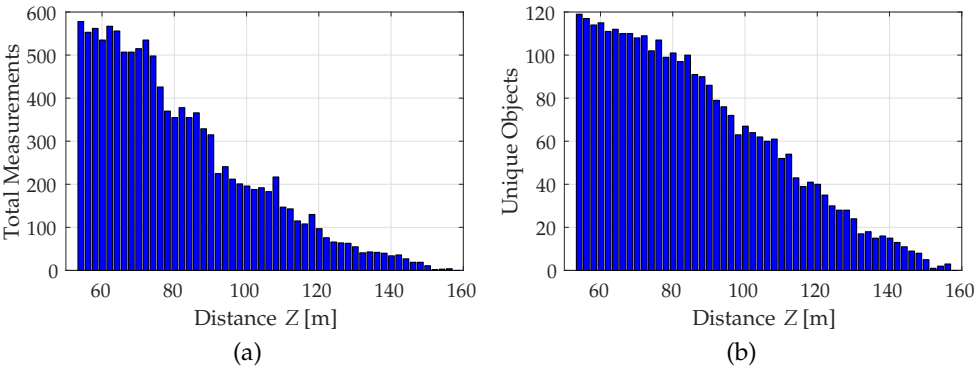


Figure 4.5: Distribution of total measurements (a) and unique observed objects (b) in the dataset.

We consider disparities between 9 and 3 pixels in the evaluation, corresponding to a distance range of approximately 50 to 160 meters. To also analyze matching accuracy as a function of absolute distance, we divide the overall range into disparity intervals of 1/20 pixel and evaluate each interval separately.

Note that, before passing the object patches to the stereo algorithms, we optimize the patch fit around objects in order to minimize the amount of outlier pixels. We exploit a precomputed dense disparity result to estimate the mean disparity for each patch and decrease the patch size until the number of outliers falls below a given threshold. Subsequently, we shrink the patches by another 25%, except for the segmentation-based approaches, where we actually increase the size again by 25%. To determine the benefit of this adapted patch fit, we also apply the LDM+ algorithm to the patches without the disparity-based patch fit optimization, denoting this variant as LDM-.

Figure 4.6: Highway driving scene with relevant objects at distances of 80 and 140 m.

4.4.3.1 *Calibration Inaccuracies*

We further include an analysis of the influence of calibration inaccuracies on the matching performance of the considered algorithms. In particular, we examine offsets in relative yaw and pitch angles, i. e. the corresponding vertical and horizontal pixel displacements, as these tend to have the most significant impact on matching performance (see Sect. 2.3.5).

4.4.4 *Results*

Table 4.2 gives an overview of the main quantitative results across the complete dataset. Fig. 4.7 shows the corresponding distributions of disparity error ϵ_d and error variation $\nabla\epsilon_d$.

Examining the overall mean $\overline{\epsilon_d}$ of the disparity error, it can be seen that the absolute values for all algorithms lie close to the ideal value of zero, within an interval of less than $1/30$ pixel. Fig. 4.8 illustrates the consistency of these results across the full distance range. While there are some variations in the mean values of the various algorithms, in practice such a small systematic disparity bias can be corrected by an online adjustment of the calibration parameters, e. g. as done in [Franke et al., 2013]. Note that the mean of $\nabla\epsilon_d$ is exactly zero for all algorithms, other values would in fact imply a temporal drift of the matching results.

Consequently, in the following we focus on the statistical variability of the measurements and mainly consider the scale estimates $S_n(\epsilon_d)$ and $S_n(\nabla\epsilon_d)$, which provide more meaningful information with regard to algorithmic matching performance.

Overall, we observe that after optimization of the selected algorithms, the differences in the results become very small. Nevertheless, it can be seen clearly that the methods which robustly combine multiple differential measurements outperform all other approaches in both disparity error and temporal error variation scales. M-LDM and M-LDM 2D obtain values as low as $S_n(\epsilon_d) = 1/13$ pixel and $S_n(\nabla\epsilon_d) = 1/20$ pixel. Notably,

Table 4.2: Overview of quantitative results, sorted by decreasing $S_n(\epsilon_d)$. See text for details.

	$S_n(\epsilon_d)$ [px]	$S_n(\nabla\epsilon_d)$ [px]	$\overline{\epsilon_d}$ [px]
SGM	0.130	0.071	−0.012
LDM-	0.120	0.065	−0.017
SGM+PLC	0.108	0.069	0.003
LDM	0.103	0.064	0.007
LDM+	0.103	0.060	0.008
MSEG	0.103	0.060	−0.010
MSEG+	0.102	0.060	−0.017
TV	0.085	0.068	0.021
TV+	0.085	0.066	−0.0091
FPHT	0.082	0.048	−0.027
M-LDM	0.076	0.048	−0.003
M-LDM 2D	0.075	0.046	−0.001

these are closely followed by FPHT, demonstrating the accuracy of the intrinsically optimized obstacle hypothesis models.

The TV and TV+ approaches also do very well in terms of $S_n(\epsilon_d)$, but perform noticeably worse with regard to temporal error variation.

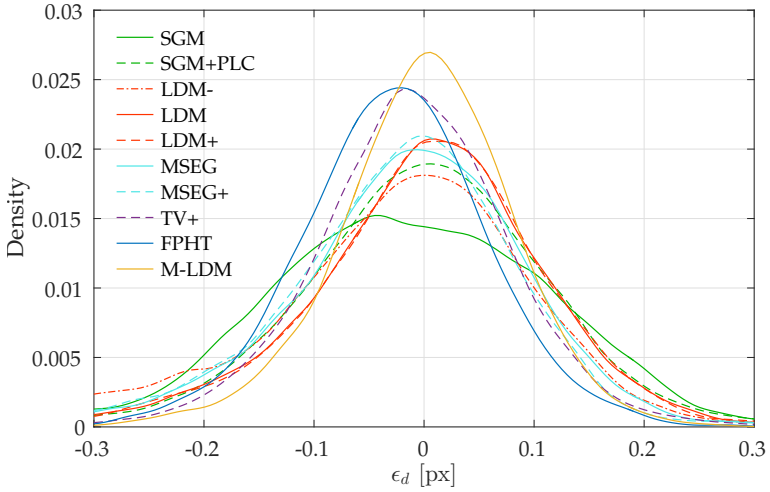
The patch-based differential matching methods LDM, LDM+, MSEG and MSEG+ all perform very similarly, yielding values of approximately $1/10$ and $1/16$ pixel for $S_n(\epsilon_d)$ and $S_n(\nabla\epsilon_d)$, respectively.

The SGM algorithm does worst of all considered methods, which is mostly due to severe pixel-locking artifacts.

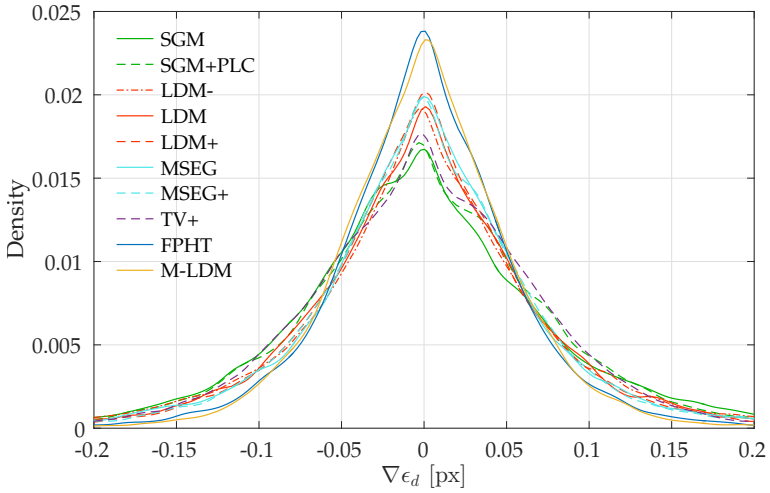
The order of the algorithms in terms of the specified performance measures is largely consistent over the whole distance range, as shown in Fig. 4.9 and Fig. 4.10. Given the properties of the used image data, the observed errors roughly agree with the results presented in [Sabater et al., 2011] on synthetic data.

4.4.4.1 *Optimized Patch Fit*

Optimizing the object patch fit has noticeable impact on the disparity error, as LDM- clearly performs worse than all other patch-based differential algorithms at $S_n(\epsilon_d)$. The error variation scale $S_n(\nabla\epsilon_d)$ without optimized patch fit is also slightly higher than in the otherwise equivalent LDM+ implementation. The efficient adaptation of the rectangular patch fit leads to a similar level of accuracy as the much more complex pixel-wise segmentation approaches MSEG and MSEG+.



(a)



(b)

Figure 4.7: Overall distributions of disparity error (a) and disparity error variation (b).

4.4.4.2 *Image Derivative Estimation and Interpolation*

Interestingly, when comparing different derivative kernels and interpolation methods, we see only very small variations in the accuracy results of the differential matching algorithms. Here, the LDM+ entry of Table 4.2 represents the default variant, with a 3×3 Scharr kernel and cubic B-Spline interpolation, whereas Table 4.3 displays the additional configurations.

Considering the derivative kernels, it appears that the 3×3 Scharr kernel performs marginally better than the alternative 5×5 variants.

With regard to the various interpolation methods, the small differences only become visible when looking at the actual sub-pixel disparity distributions in Fig. 4.11. Cubic B-Spline interpolation produces a nearly uniform distribution, while cubic convolution and bilinear interpolation re-

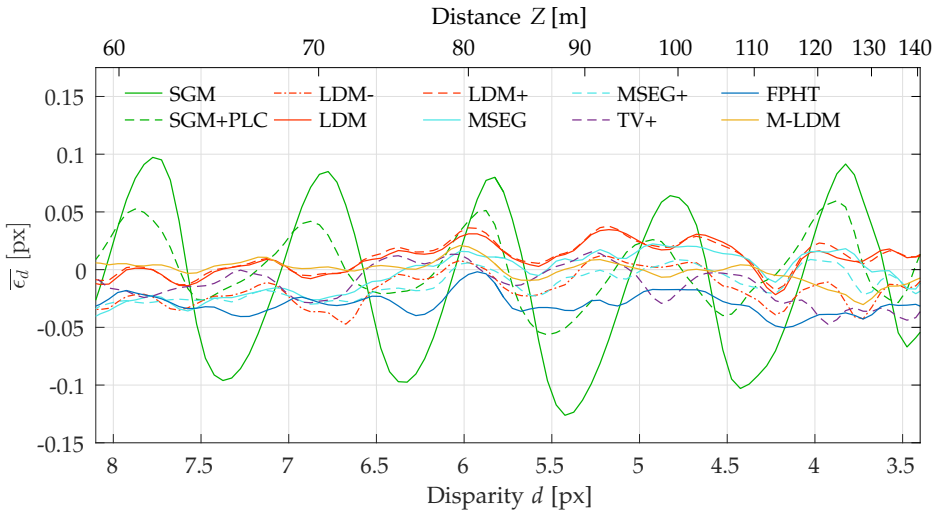


Figure 4.8: Mean of disparity error over distance range.

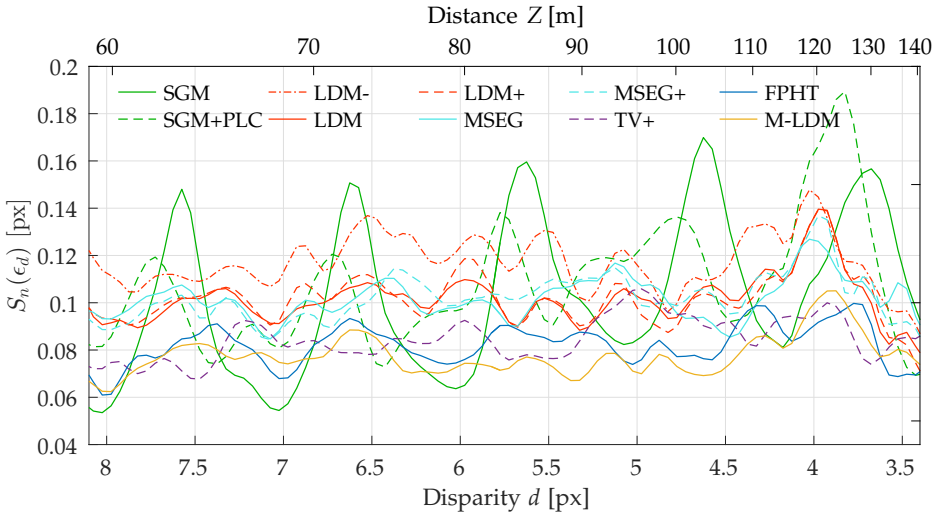


Figure 4.9: Scale estimate of disparity error over distance range.

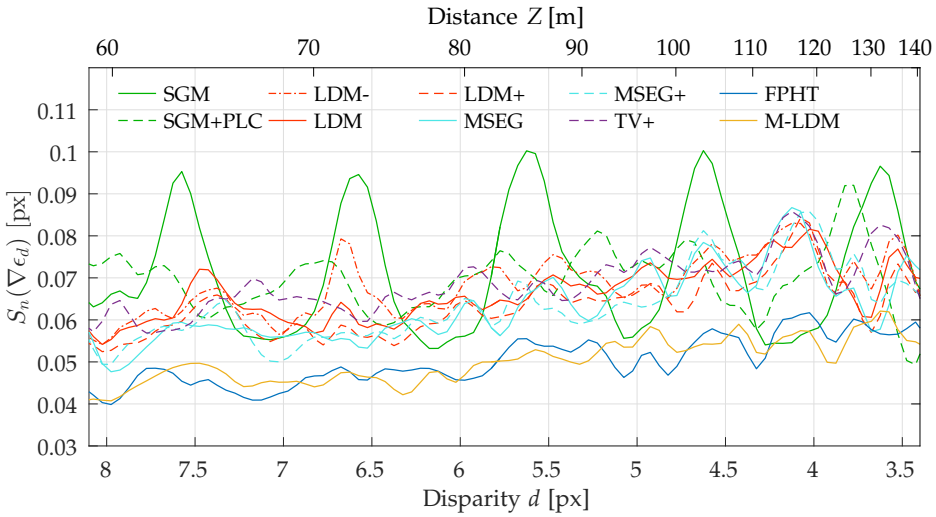


Figure 4.10: Scale estimate of disparity error variation over distance range.

Table 4.3: Impact of derivative kernels and interpolation methods on LDM+ results.

	$S_n(\epsilon_d)$ [px]	$S_n(\nabla\epsilon_d)$ [px]
Scharr 3×3, B-Spline	0.103	0.060
Scharr 5×5	0.110	0.065
Central Differences 5×5	0.109	0.064
Bilinear	0.106	0.062
Cubic Convolution	0.103	0.061

sult in a very slight bias towards half pixels. These small variations are in agreement with theoretical predictions presented in [Sutton et al., 2009], but are not distinguishable by our practical disparity accuracy measures at this scale.

4.4.4.3 *Pixel-Locking Compensation*

In contrast, the systematic pixel-locking effect of the census-based SGM algorithm is clearly visible in both the disparity sub-pixel distribution of Fig. 4.11a and in the error measures analyzed over absolute distance (Fig. 4.8, Fig. 4.9, Fig. 4.10). As a consequence of the pixel-locking effect, SGM fares worst of all algorithms in overall $S_n(\epsilon_d)$ and $S_n(\nabla\epsilon_d)$ error scores. However, applying the proposed compensation method considerably reduces the effect, bringing SGM+PLC closer to the performance of the differential matching algorithms. More sophisticated compensation approaches can be expected to reduce this gap even further.

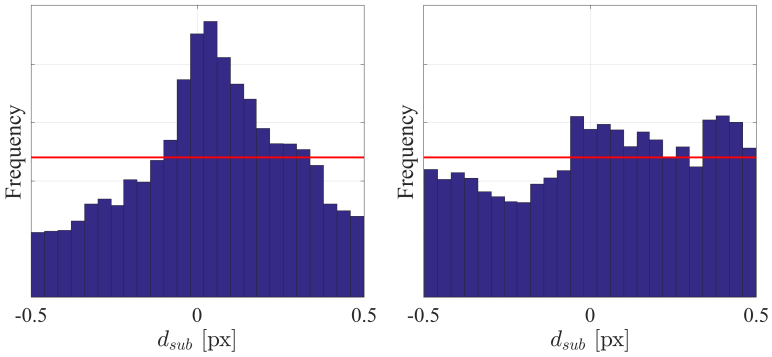
Looking at the sub-pixel distribution of the FPHT disparity estimates in Fig. 4.11g, we can see a tiny peak at integer disparities. This is likely due to the initialization of the FPHT obstacle models via SGM, as the subsequent optimization procedure sometimes only results in a rotation of the local plane models.

4.4.4.4 *Symmetric Residuals*

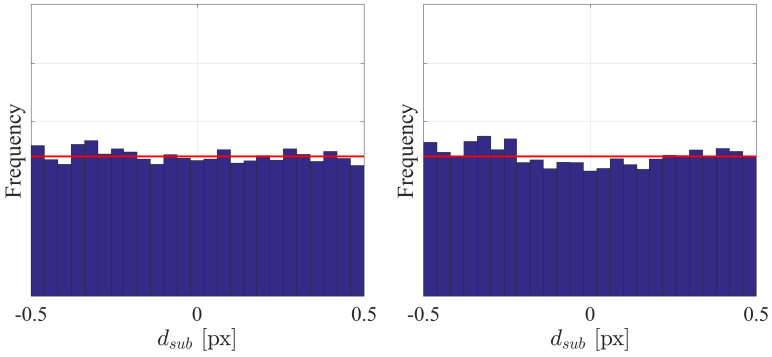
Now we examine the impact of utilizing symmetric residuals in LDM+ and TV+. As can be seen from Table 4.2, Fig. 4.7a and Fig. 4.9, LDM+ yields the same results as LDM for $S_n(\epsilon_d)$, but performs slightly better in terms of error variation. TV and TV+ achieve virtually identical results, the global regularization effectively neutralizing the small differences in data terms.

4.4.4.5 *Scene Flow*

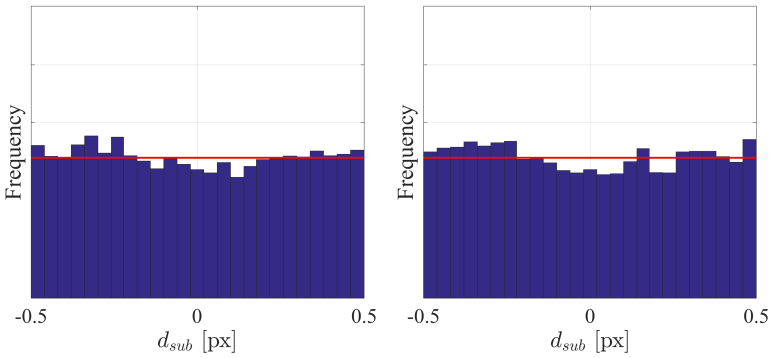
Finally, our evaluation shows that utilizing the data of two consecutive stereo pairs for scene flow segmentation and disparity refinement as in MSEG+ does not necessarily yield a measurable improvement in terms



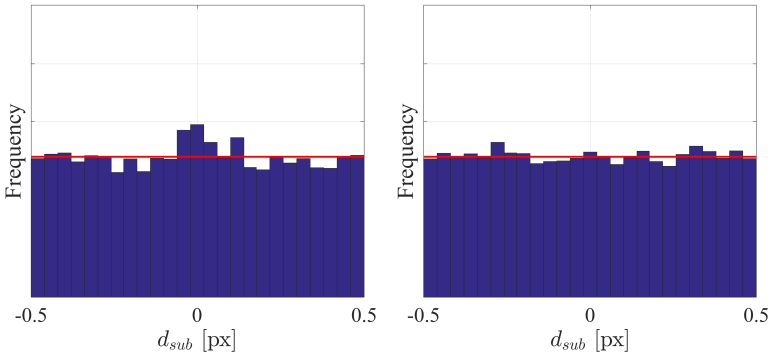
(a) SGM: V-fit (b) SGM+PLC: V-fit



(c) LDM+: B-Spline (d) LDM+: Cubic Conv.



(e) LDM+: Bilinear (f) TV+: Bilinear



(g) FPHT (h) M-LDM

Figure 4.11: Sub-pixel disparity distributions resulting from different matching and interpolation methods. Plots show the interval $[-0.5, 0.5]$ centered on full pixel disparities.

of matching accuracy. This might be due to the fact that, in order to align all images, two additional sets of two-dimensional displacements have to be estimated, introducing errors not present in the standard two-image computation. Also, applying a more sophisticated imaging model for estimating the unknown signal could further improve results.

4.4.4.6 *Calibration Inaccuracies*

After analyzing the stereo matching performance under optimized calibration parameters, we now examine the algorithms' robustness to calibration errors. We separately consider inaccuracies in the relative yaw and pitch angles of the two cameras, resulting in horizontal and vertical displacements of the rectified images, respectively. The left column of Fig. 4.12 illustrates the impact of horizontal offsets on $\overline{\epsilon_d}$, $S_n(\epsilon_d)$ and $S_n(\nabla\epsilon_d)$, whereas the right column shows the impact of vertical offsets.

As expected, the mean disparity error of all algorithms increases proportionally to the magnitude of the horizontal offset (Fig. 4.12a), whereas $S_n(\epsilon_d)$ (Fig. 4.12c) and $S_n(\nabla\epsilon_d)$ (Fig. 4.12e) remain virtually unaffected.

In contrast, a vertical offset between the stereo images leaves $\overline{\epsilon_d}$ almost unchanged (Fig. 4.12b), but has considerable impact on both $S_n(\epsilon_d)$ (Fig. 4.12d) and $S_n(\nabla\epsilon_d)$ (Fig. 4.12f). The SGM algorithm proves to be quite robust to the vertical sub-pixel offsets, owing to the used census transform and the relatively large matching windows. This is in good agreement with results reported by Hirschmüller and Gehrig [2009]. The patch-based differential matching methods (LDM, LDM+, MSEG) do slightly worse than SGM here, but also benefit from their relatively large matching windows. The approaches combining multiple independent measurements (M-LDM, FPHT), along with TV, show the strongest increase in error scales, once the offsets go beyond 0.2 pixels. However, by explicitly estimating the vertical offset in conjunction with the disparity, these drawbacks can largely be eliminated. This concept makes the M-LDM 2D approach effectively invariant to sub-pixel errors caused by relative pitch angle inaccuracies.

Overall, the above observations once more highlight the importance of accurate estimation and maintenance of calibration parameters in practice.

4.4.4.7 *Runtimes*

Without prior knowledge of object locations, in a first step the top-performing methods M-LDM 2D, M-LDM and FPHT are applied to compute independent measurements over the full input images. For FPHT, the corresponding runtimes are analyzed in Sect. 3.5.5.4. Similar to FPHT, the M-LDM measurements are computed in parallel on a GPU, taking less than 20 ms per image. Of course, these values can be reduced significantly by restricting computation to object-based regions of interest only. Once the point-wise measurements are available,

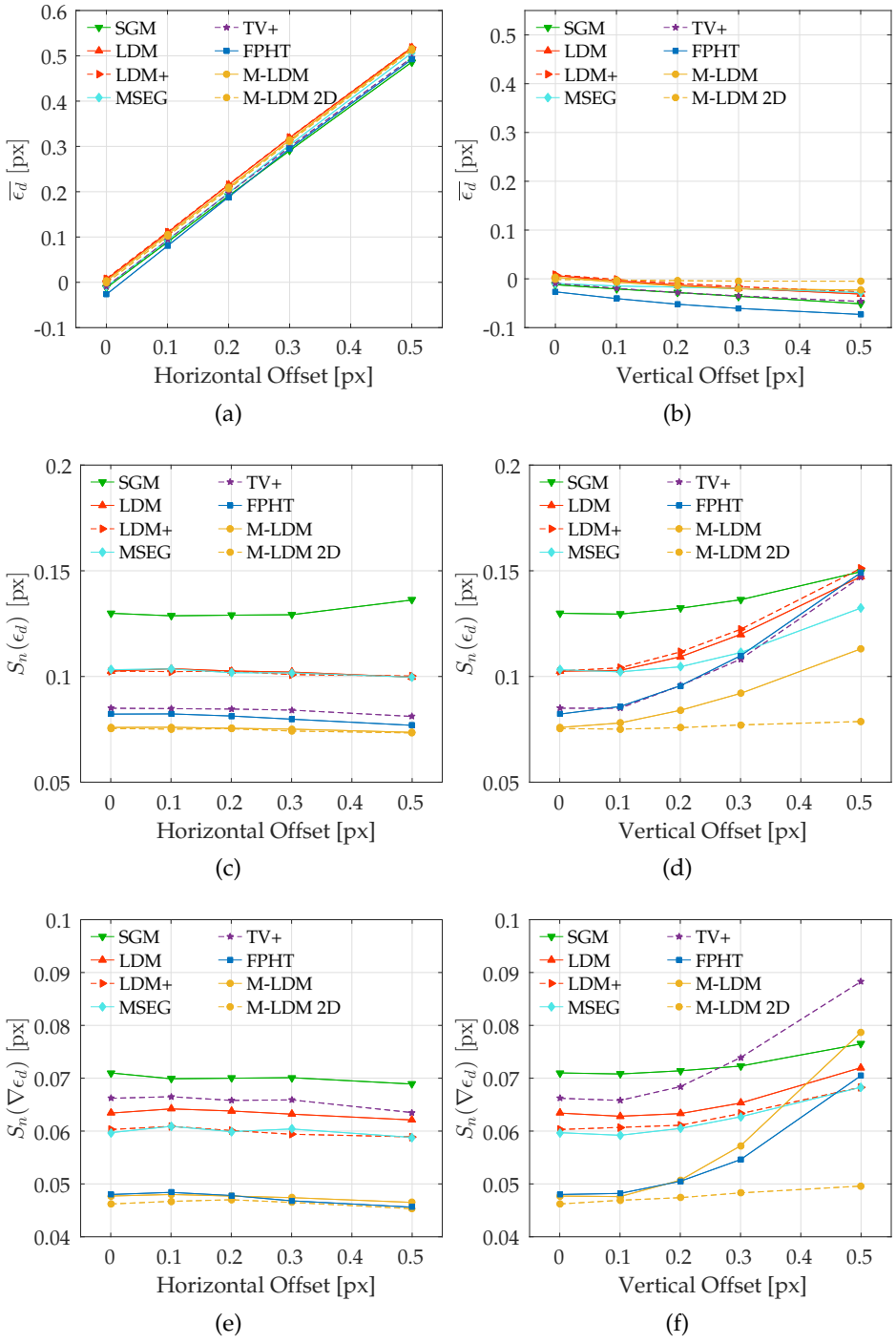


Figure 4.12: Stereo matching errors due to calibration inaccuracies.

the time taken for the combination into a single object observation is negligible.

For the global algorithms SGM and TV we make use of custom implementations on a FPGA [Gehrig et al., 2009] and a GPU [Rabe, 2011]. Here we obtain runtimes of approximately 40 ms and 65 ms, respectively. Again, the time to compute object disparities fom the dense disparity maps is negligible.

The LDM versions are initialized with the SGM output and take less than one millisecond per object, varying only insignificantly for the considered patch sizes. The more complex approaches MSEG and MSEG+ take up to 40 ms and 80 ms, respectively, as they include an additional outer iteration loop for segmentation and require a significant amount of time for graph-cut based pixel labeling, even when using the speed-up methods of Alahari et al. [2010].

4.5 SUMMARY

In this chapter we depart from the common setting of major dense stereo benchmarks and examine the sub-pixel matching accuracy for isolated salient objects. This is motivated by modern safety-relevant applications of stereo vision, where highest sub-pixel accuracy is required in selected image areas. The presented analysis is based on an extensive real-world dataset, enabling meaningful statistical evaluation and providing valuable insights regarding the matching accuracy achievable in practice. We propose the use of robust statistical measures of scale to evaluate matching performance, and additionally introduce an object-based temporal disparity error variation measure which is invariant to systematic disparity offsets.

Careful optimization of each considered stereo algorithm minimizes the observable differences in matching accuracy and yields consistent disparity error scale estimates of less than 1/10 pixel. Approaches which robustly combine multiple differential measurements outperform all other methods in both disparity error and temporal error variation scales. Notably, the obstacle point distance estimates provided by the FPHT detection approach proposed in Chapter 3 can directly be utilized for this purpose, resulting in highly accurate results on par with the top-performing dedicated matching algorithms. However, these types of methods prove to be vulnerable to calibration inaccuracies, which can be alleviated by reliable online self-calibration algorithms or the simultaneous estimation of offsets resulting from calibration errors.

The choice of derivative filter and interpolation method does not have a significant impact on the disparity accuracy of patch-based differential matching methods, while optimized patch shapes are crucial. Utilizing the full data of two consecutive stereo pairs does not necessarily yield the expected benefits, but shows potential for use with more sophisticated imaging and estimation models.

The pixel-locking effects of discrete matching methods such as SGM cause significant errors in sub-pixel disparity, but can efficiently be alleviated by dedicated correction approaches. This brings discrete methods close to differential matching algorithms in terms of accuracy.

5 CONCLUSION AND OUTLOOK

CONTENTS

In this thesis, two major challenges of environment perception systems for intelligent vehicles were addressed by means of a stereo camera system: The high-sensitivity detection of generic obstacles and the high-accuracy estimation of corresponding object distances.

5.1 OBJECT DETECTION

In Chapter 3, a novel method for high-sensitivity generic obstacle detection called Direct Planar Hypothesis Testing (PHT) was presented and analyzed in detail. The detection approach is based on a careful analysis of real-world requirements, derived from the need of intelligent vehicles to cope with all types of obstacles on all types of paved road surface geometries. In particular, this includes the detection of previously unseen as well as distant or very small obstacles.

The proposed algorithm performs pixel-wise binary hypothesis tests directly on stereo image data, assessing free-space and obstacle hypotheses on independent local patches. The tests are based on constrained, locally planar geometric hypothesis models, which provide the necessary flexibility to handle globally non-flat ground surfaces. Since the detection algorithm implicitly performs an optimization of the underlying geometric hypothesis models, the refined distance estimates for detected objects are provided as an additional output.

Extending upon the idea of PHT, in Sect. 3.3 a reparametrization of the underlying optimization problem was proposed, yielding a slightly less flexible but computationally more efficient variant called Fast Direct Planar Hypothesis Testing (FPHT). Both PHT and FPHT benefit significantly from massive parallelization, which was demonstrated by real-time execution on a GPU. An implementation using low-power dedicated hardware such as FPGAs is conceivable.

The proposed approaches yield a pixel-wise obstacle detection result, however, the corresponding raw 3D obstacle point clouds might not be an optimal input representation for subsequent processing steps. Therefore, a compact yet flexible mid-level representation called Cluster-Stixels (CStix) was presented, which was inspired by the established *Stixel World*

of Badino et al. [2009], Pfeiffer and Franke [2011] and Schneider et al. [2016]. In practical experiments, the CStix representation was shown to provide a very suitable compact description of cluttered urban traffic scenes and arbitrarily shaped obstacle types. Furthermore, an alternative Bounding Box (BB) representation of the detection output was presented and shown to serve as an appropriate input for model-based object tracking algorithms.

The proposed detection system was thoroughly evaluated and compared to a set of established baselines with a focus on two critical scenarios: The detection of small, generic obstacles in complex urban environments, and the detection of generic objects at long range, e. g. on highways. In all tests PHT/FPHT and the corresponding CStix and bounding box representations significantly outperformed the selected baselines, providing a considerable increase in detection range while reducing the false positive rate at the same time.

Despite the convincing performance of the methods presented in this work, there is still urgent need to improve upon both detection performance as well as false positive rates in order to close the gap between semi-autonomous and fully-autonomous driving functionalities. A promising extension of the present work is the application of the multi-view formulation described in Sect. 3.2.6. For example, the use of an L-shaped trinocular stereo setup as sketched in Fig. 5.1 can be expected to result in a considerable performance improvement. Such a setup facilitates the use of image texture in both vertical and horizontal directions to support obstacle decisions, and to help resolve ambiguities during the optimization of hypothesis models. By using an additional data source the influence of noise and outliers can be reduced more effectively. The orthogonal configuration would even allow for a minimal parameterization of the detection problem as used in FPHT.

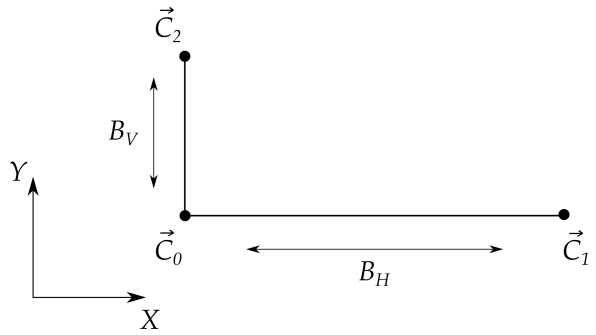


Figure 5.1: A trinocular stereo configuration with cameras located at $\vec{C}_0$, $\vec{C}_1$ and $\vec{C}_2$ can be expected to yield a considerable performance improvement over the standard stereo cameras used in this work. The setup allows to exploit the vertical displacement B_V in addition to the horizontal displacement B_H.

A different, very promising direction of future work is the combination of geometric detection approaches with appearance-based state-of-the-art machine learning methods as recently shown in [Ramos et al., 2017]. A principled fusion concept holds the potential to boost detection rates even further while significantly reducing false positives at the same time.

In practice, visual perception systems of autonomous vehicles would utilize the proposed specialized object detection methods to augment existing general-purpose 3D scene representation approaches such as the semantic *Stixel World* of Schneider et al. [2016]. In this way, a holistic scene representation is obtained, providing the vehicle with a comprehensive understanding of its environment while being able to handle the challenging cases considered in this thesis.

5.2 DISTANCE ESTIMATION

In Chapter 4, the task of object distance estimation from stereo vision was analyzed in detail. As obstacles along the path of motion are especially relevant for collision avoidance, their location and velocity have to be determined with maximum accuracy. Hence, highest sub-pixel stereo matching performance in selected image areas is required. This stands in contrast to the common setting of major stereo benchmarks, where average matching performance across full images is evaluated.

In this work, several new approaches for optimizing the stereo matching accuracy for isolated salient objects were proposed and compared to state-of-the-art concepts. The set of algorithms includes Local Differential Matching (LDM) methods as well as Joint Matching and Segmentation (MSEG), but also approaches performing global optimization of pixel-wise costs in both discrete and continuous settings. Moreover, the robust combination of multiple independent, local observations into a single optimized object distance estimate was considered.

To allow for a meaningful statistical analysis and a systematic assessment of matching errors, this work proposed the use of robust, location-free measures of scale, along with a novel object-based temporal disparity error variation measure. The study was performed on a large dedicated dataset, using a long-range RADAR sensor for reference.

Overall, the best accuracy was obtained by the robust combination of multiple independent observations per object. Notably, the obstacle point distance estimates provided by the proposed PHT/FPHT detection system can directly be utilized for this purpose, resulting in highly accurate results on par with the top-performing dedicated matching algorithms. The attained error scale estimates lie below 1/10 pixel, with a temporal error variation of less than 1/20 pixel. Across all considered algorithms, the largest matching errors were caused by the use of improper object support or by the pixel-locking artifacts of discrete matching methods.

In addition to providing reference values for the matching accuracy achievable in practice, the presented experiments also illustrate the influence of calibration errors, which highlights the necessity of reliable online self-calibration algorithms.

In future experiments, state-of-the-art long-range LIDAR sensors could be employed to obtain ground truth data of even better quality, regarding both accuracy and density.

Naturally, the analysis could be extended further by including additional stereo algorithms, for example variants of phase correlation-based local matching. As an alternative to the presented Joint Matching and Segmentation (MSEG) algorithms, recently proposed machine learning-based methods such as Mask R-CNN [He et al., 2017] offer a powerful tool for optimizing the object support used for matching. Such methods could also be used to alleviate potential foreground fattening artifacts resulting from the PHT/FPHT object detection algorithms. Beyond that, recent developments towards direct regression of stereo disparity via Convolutional Neural Networks (CNNs), as e. g. shown by Kendall et al. [2017], open up a new field of possibilities to be considered in subsequent studies.

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

BIBLIOGRAPHY

ADAC. Zahlen, Fakten, Wissen. Aktuelles aus dem Verkehr. Technical report, 2016. URL https://www.adac.de. (Cited on page 1.)

K. Alahari, P. Kohli, and P. H. S. Torr. Dynamic Hybrid Algorithms for MAP Inference in Discrete MRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(10):1846–57, 2010. (Cited on page 115.)

H. Badino, U. Franke, and D. Pfeiffer. The Stixel World - A Compact Medium Level Representation of the 3D-World. In *DAGM Symposium*, 2009. (Cited on pages viii, 32, 34, and 118.)

S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework: Part 1. *International Journal of Computer Vision (IJCV)*, 56(3):221–255, 2004. (Cited on pages 51, 55, and 87.)

S. Baker, R. Gross, and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework: Part 4. Technical Report CMU-RI-TR-04-14, Carnegie Mellon University, 2004. (Cited on page 98.)

S. Baker, A. Datta, and T. Kanade. Parameterizing homographies. Technical Report CMU-RI-TR-06-11, Carnegie Mellon University, 2006. (Cited on page 41.)

N. Bernini, M. Bertozzi, L. Castangia, M. Patander, and M. Sabbatelli. Real-Time Obstacle Detection using Stereo Vision for Autonomous Ground Vehicles: A Survey. In *IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014. (Cited on page 32.)

M. Bertozzi, L. Bombini, A. Broggi, M. Buzzoni, E. Cardarelli, S. Cattani, P. Cerri, A. Coati, S. Debattisti, A. Falzoni, R. I. Fedriga, M. Felisa, L. Gatti, A. Giacomazzo, P. Grisleri, M. C. Laghi, L. Mazzei, P. Medici, M. Panciroli, P. P. Porta, P. Zani, and P. Versari. VIAC: An Out of Ordinary Experiment. In *IEEE Intelligent Vehicles Symposium (IV)*, 2011. (Cited on page 2.)

J. Besag. Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society*, 36(2):192–236, 1974. (Cited on page 94.)

R. Bichsel and P. Borges. Low-Obstacle Detection Using Stereo Vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016. (Cited on page 32.)

S. D. Blostein and T. S. Huang. Error Analysis in Stereo Determination of 3-D Point Positions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 9(6), 1987. (Cited on pages 28 and 29.)

J. Bouguet. Camera Calibration Toolbox for Matlab, 2017. URL http://www.vision.caltech.edu/bouguetj/calib{_}doc/. (Cited on pages 22 and 23.)

Y. Boykov and G. Funka-Lea. Graph Cuts and Efficient N-D Image Segmentation. *International Journal of Computer Vision (IJCV)*, 70(2):109–131, nov 2006. (Cited on page 95.)

Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(9): 1124–1137, 2004. (Cited on page 95.)

Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(11):1222–1239, 2001. (Cited on page 95.)

H. H. Braess and G. Reichart. Prometheus: Vision des 'intelligenten Automobils' auf 'intelligenter Straße'? Versuch einer kritischen Würdigung - Teil 1. *ATZ Automobiltechnische Zeitschrift*, 97(4), 1995a. (Cited on page 2.)

H. H. Braess and G. Reichart. Prometheus: Vision des 'intelligenten Automobils' auf 'intelligenter Straße'? Versuch einer kritischen Würdigung - Teil 2. *ATZ Automobiltechnische Zeitschrift*, 97(6), 1995b. (Cited on page 2.)

A. Broggi, M. Buzzoni, M. Felisa, and P. Zani. Stereo Obstacle Detection in Challenging Environments: The VIAC Experience. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1599–1604, 2011. (Cited on pages 32, 34, 57, 68, and 121.)

D. C. Brown. Decentering Distortion of Lenses. *Photogrammetric Engineering*, 32(3):444–462, 1966. (Cited on page 21.)

D. C. Brown. Close-Range Camera Calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971. (Cited on page 21.)

C. Chang, S. Chatterjee, and P. R. Kube. A Quantization Error Analysis for Convergent Stereo. In *IEEE International Conference on Image Processing (ICIP)*, 1994. (Cited on page 28.)

M. Cordts, L. Schneider, U. Franke, and S. Roth. Object-level Priors for Stixel Generation. In *German Conference on Pattern Recognition (GCPR)*, 2014. (Cited on pages 33, 65, and 67.)

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. (Cited on pages 33 and 65.)

D. Cremers and A. Yuille. A Generative Model Based Approach to Motion Segmentation. In *DAGM Symposium*, 2003. (Cited on page 91.)

C. Creusot and A. Munawar. Real-time small obstacle detection on highways using compressive RBM road reconstruction. In *IEEE Intelligent Vehicles Symposium (IV)*, 2015. (Cited on page 33.)

T. Dang, C. Hoffmann, and C. Stiller. Continuous Stereo Self- Calibration by Camera Parameter Tracking. *IEEE Transactions on Image Processing (TIP)*, 18(7):1536–50, jul 2009. (Cited on page 27.)

E. D. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrandt, M. Maurer, F. Thomanek, and J. Schiehlen. The Seeing Passenger Car 'VaMoRs-P'. In *IEEE Intelligent Vehicles Symposium (IV)*, 1994. (Cited on page 2.)

M. Elad, P. Teo, and Y. Hel-Or. On the Design of Filters for Gradient-Based Motion Estimation. *Journal of Mathematical Imaging and Vision*, 23(3):345–365, 2005. (Cited on pages 85 and 89.)

M. Enzweiler and D. M. Gavrila. Monocular Pedestrian Detection: Survey and Experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(12):2179–2195, 2009. (Cited on page 33.)

M. Enzweiler and D. M. Gavrila. A Multilevel Mixture-of-Experts Framework for Pedestrian Classification. *IEEE Transactions on Image Processing (TIP)*, 20(10):2967–2979, 2011. (Cited on page 105.)

M. Enzweiler, M. Hummel, D. Pfeiffer, and U. Franke. Efficient Stixel-Based Object Recognition. In *IEEE Intelligent Vehicles Symposium (IV)*, 2012. (Cited on page 33.)

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *International Conference on Knowledge Discovery and Data Mining*, 1996. (Cited on page 57.)

M. Fach and D. Ockel. Evaluation Methods for the Effectiveness of Active Safety Systems with Respect to Real World Accident Analysis. In *International Technical Conference on the Enhanced Safety of Vehicles*, 2009. (Cited on page 1.)

H. Farid and E. P. Simoncelli. Differentiation of Discrete Multidimensional Signals. *IEEE Transactions on Image Processing (TIP)*, 13(4):496–508, apr 2004. (Cited on pages 85 and 89.)

O. Faugeras and Q.-T. Luong. *The Geometry of Multiple Images: The Laws that Govern the Formation of Multiple Images of a Scene and Some of their Applications*. MIT Press, 2001. (Cited on pages 19 and 22.)

F. Fooladgar, S. Samavi, S. M. R. Soroushmehr, and S. Shirani. Geometrical Analysis of Localization Error in Stereo Vision Systems. *IEEE Sensors Journal*, 13(11), 2013. (Cited on page 28.)

W. Förstner. Image Matching. In R. M. Haralick and L. G. Shapiro, editors, *Computer and Robot Vision*, chapter 16, pages 289–372. Addison-Wesley, 2. edition, 1993. (Cited on page 85.)

D. Forsyth and J. Ponce. *Computer Vision - A Modern Approach*. Pearson, 2002. (Cited on page 19.)

P. E. Frandsen, K. Jonasson, H. B. Nielsen, and O. Tingleff. Unconstrained Optimization. Technical report, Technical University of Denmark, 2004. (Cited on pages 7 and 9.)

U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, and R. G. Herrtwich. Making Bertha See. In *International Conference on Computer Vision Workshops (ICCVW)*, 2013. (Cited on pages 2, 35, 105, and 107.)

C. Freundlich, M. Zavlanos, and P. Mordohai. Exact Bias Correction and Covariance Estimation for Stereo Vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. (Cited on page 28.)

A. Fusiello, E. Trucco, and A. Verri. A Compact Algorithm for Rectification of Stereo Pairs. *Machine Vision and Applications*, (March):16–22, 2000. (Cited on page 23.)

S. K. Gehrig and U. Franke. Improving Stereo Sub-Pixel Accuracy for Long Range Stereo. In *International Conference on Computer Vision Workshops (ICCVW)*, 2007. (Cited on page 85.)

S. K. Gehrig, F. Eberli, and T. Meyer. A Real-Time Low-Power Stereo Vision Engine Using Semi-Global Matching. In *ICVS*, 2009. (Cited on pages 102 and 115.)

S. K. Gehrig, H. Badino, and U. Franke. Improving Stereo Sub-Pixel Accuracy for Long Range Stereo. *Computer Vision and Image Understanding (CVIU)*, 116(1):16–24, 2012. (Cited on pages 91 and 123.)

S. K. Gehrig, R. Stalder, and N. Schneider. A Flexible High-Resolution Real-Time Low-Power Stereo Vision Engine. In *International Conference on Computer Vision Systems (ICVS)*, 2015. (Cited on pages 25, 70, 96, 102, and 121.)

A. Geiger, P. Lenz, and R. Urtasun. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. ISBN 978-1-4673-1228-8. doi: 10.1109/CVPR.2012.6248074. (Cited on pages 25, 26, 28, 83, and 84.)

K. Granström, M. Baum, and S. Reuter. Extended Object Tracking: Introduction, Overview and Applications. *Journal of Advances in Information Fusion (ISIF)*, 12(2), 2017. (Cited on page 61.)

E. Guizzo. How Google's Self-Driving Car Works. *IEEE Spectrum Online*, 18, 2011. (Cited on page 2.)

A. Guttman. R-trees: A Dynamic Index Structure for Spatial Searching. In *ACM SIGMOD*, pages 47–57, 1984. (Cited on page 57.)

R. Hadsell, A. Erkan, P. Sermanet, and M. Scoffier. Deep Belief Net Learning in a Long-Range Vision System for Autonomous Off-Road Driving. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008. (Cited on page 32.)

G. D. Hager and P. N. Belhumeur. Efficient Region Tracking With Parametric Models of Geometry and Illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(10):1025–1039, 1998. (Cited on page 55.)

I. Haller and S. Nedevschi. Design of Interpolation Functions for Subpixel-Accuracy Stereo-Vision Systems. *IEEE Transactions on Image Processing (TIP)*, 21(2):889–98, feb 2012. (Cited on pages 28, 85, 102, and 103.)

J. Hammersley and P. Clifford. Markov Fields on Finite Graphs and Lattices. Technical report, 1971. (Cited on page 94.)

A. Harakeh, D. Asmar, and E. Shammas. Ground Segmentation and Occupancy Grid Generation Using Probability Fields. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015. (Cited on page 32.)

R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2nd edition, 2004. (Cited on pages 14, 15, 19, 20, 22, and 40.)

R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding (CVIU)*, 1997. (Cited on page 24.)

K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. In *International Conference on Computer Vision (ICCV)*, 2017. (Cited on page 120.)

J. Heikkilä and O. Silven. A Four-Step Camera Calibration Procedure with Implicit Image Correction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1106–1112. IEEE, 1997. (Cited on pages 22 and 23.)

H. Hirschmüller. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):328–41, feb 2008. ISSN 0162-8828. (Cited on pages 25, 102, and 121.)

H. Hirschmüller and S. Gehrig. Stereo Matching in the Presence of Sub-Pixel Calibration Errors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. (Cited on pages 27 and 113.)

P. J. Huber. Robust Estimation of a Location Parameter. *Annals of Mathematical Statistics*, 35(1), 1964. (Cited on pages 14 and 15.)

P. J. Huber and E. M. Ronchetti. *Robust Statistics*. Wiley, 2nd edition, 2009. (Cited on page 14.)

K. Iagnemma, R. Kurjanowicz, and M. Buehler. Special Issue on the DARPA Grand Challenge - Part 1. *Journal of Field Robotics*, 23(8), 2006a. (Cited on page 2.)

K. Iagnemma, R. Kurjanowicz, and M. Buehler. Special Issue on the DARPA Grand Challenge - Part 2. *Journal of Field Robotics*, 23(9), 2006b. (Cited on page 2.)

B. Jähne. *Digital Image Processing - Concepts, Algorithms, and Scientific Applications*. Springer, 3rd edition, 1995. (Cited on page 89.)

O. Kähler and J. Denzler. Tracking and Reconstruction in a Combined Optimization Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(2):387–401, 2012. (Cited on page 43.)

S. Kammel, J. Ziegler, B. Pitzer, T. Gindele, D. Jagzent, J. Schr, and F. V. Hundelshausen. Team AnnieWAY ' s Autonomous System for the 2007 DARPA Urban Challenge. *Journal of Field Robotics*, 25(9):615–639, 2008. (Cited on page 2.)

C. Kanzow, N. Yamashita, and M. Fukushima. Levenberg-Marquardt Methods with Strong Local Convergence Properties for Solving Nonlinear Equations with Convex Constraints. *Journal of Computational and Applied Mathematics*, 172:375–397, 2004. (Cited on page 12.)

S. M. Kay. *Fundamentals of Statistical Signal Processing: Detection Theory*, volume II. Prentice Hall, 1998. (Cited on pages 16, 17, and 18.)

A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-End Learning of Geometry and Context for Deep Stereo Regression. In *International Conference on Computer Vision (ICCV)*, 2017. (Cited on pages 26, 84, and 120.)

R. G. Keys. Cubic Convolution Interpolation for Digital Image Processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6): 1153–1160, 1981. (Cited on page 89.)

S. Kramm and A. Bensrhair. Obstacle Detection Using Sparse Stereovision and Clustering Techniques. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 760–765, 2012. (Cited on page 32.)

R. Labayrade, D. Aubert, and J.-P. Tarel. Real Time Obstacle Detection in Stereovision on Non Flat Road Geometry Through "V-disparity". In *IEEE Intelligent Vehicles Symposium (IV)*, 2002. (Cited on pages 32 and 49.)

S. Leutenegger, M. Lopez, and J. Edgington. STR: A Simple and Efficient Algorithm for R-tree Packing. In *International Conference on Data Engineering*, pages 497–506, 1997. (Cited on page 57.)

J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards Fully Autonomous Driving: Systems and Algorithms. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168, 2011. (Cited on page 2.)

S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Advances in Pattern Recognition. Springer, 3rd edition, 2010. (Cited on page 94.)

A. Lie, C. Tingvall, M. Krafft, and A. Kullgren. The Effectiveness of Electronic Stability Control (ESC) in Reducing Real Life Crashes and Injuries. *Traffic Injury Prevention*, 7(1):38–43, 2006. (Cited on page 1.)

T. Litman. Autonomous Vehicle Implementation Predictions: Implications for Transport Planning. *Transportation Research Board Annual Meeting*, 2015. (Cited on page 2.)

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision (ECCV)*, 2016. (Cited on page 33.)

J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. (Cited on page 33.)

M. Lourakis and S. Orphanoudakis. Visual Detection of Obstacles Assuming a Locally Planar Ground. In *Asian Conference on Computer Vision (ACCV)*, 1998. (Cited on page 32.)

H. Lu, L. Jiang, and A. Zell. Long Range Traversable Region Detection Based on Superpixels Clustering for Mobile Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015. (Cited on page 33.)

B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *International Joint Conference on Artificial Intelligence*, 1981. (Cited on pages 44, 85, and 86.)

J. M. Lutin, A. L. Kornhauser, and E. Lerner-Lam. The Revolutionary Development of Self-Driving Vehicles and Implications for the Transportation Engineering Profession. *Institute of Transportation Engineers (ITE) Journal*, (July):5, 2013. (Cited on page 2.)

K. Madsen, H. Nielsen, and O. Tingleff. Methods for Non-Linear Least Squares Problems. Technical report, Technical University of Denmark, 2004. (Cited on pages 7, 8, 11, and 12.)

R. Manduchi, A. Castano, A. Talukder, and L. Matthies. Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation. *Autonomous Robots*, 18:81–102, 2005. (Cited on pages 32, 34, 57, 68, 74, 76, and 121.)

L. Matthies and S. A. Shafer. Error Modelling in Stereo Navigation. *IEEE Journal of Robotics and Automation*, 3(3), 1987. (Cited on page 28.)

N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. (Cited on page 26.)

J. C. McGlone, E. M. Mikhail, and J. S. Bethel, editors. *Manual of Photogrammetry*. ASPRS, 5th edition, 2004. (Cited on page 21.)

M. Menze and A. Geiger. Object Scene Flow for Autonomous Vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. (Cited on pages 25, 26, 28, 83, and 84.)

R. Mester. Motion Estimation Revisited: An Estimation-Theoretic Approach. In *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, 2014. (Cited on page 44.)

I. Miller and M. Campbell. Team Cornell's Skynet: Robust Perception and Planning in an Urban Environment. *Journal of Field Robotics*, 25(8): 493–527, 2008. (Cited on page 2.)

M. Montemerlo, H. Dahlkamp, D. Dolgov, D. Haehnel, G. Hoffmann, D. Johnston, A. Levandowski, J. Levinson, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, D. Stavens, A. Vogt, S. Thrun, J. Becker, S. Shat, T. Hilden, B. Huhnke, S. Klumpp, D. Langer, J. Marcil, M. Pflueger, G. Stanek, and S. Ettinger. Junior: The Stanford Entry in the Urban Challenge. *Journal of Field Robotics*, 25(9):569–597, 2008. (Cited on page 2.)

S. Nedevschi, T. Marita, R. Danescu, F. Oniga, D. Frentiu, and C. Pocol. Camera Calibration Error Analysis. In *microCAD*, 2003. (Cited on page 27.)

S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, T. Graf, and R. Schmidt. High Accuracy Stereovision Approach for Obstacle Detection on Non-Planar Roads. *IEEE Inteligent Engineering Systems (INES)*, 2004a. (Cited on pages 32 and 58.)

S. Nedevschi, R. Schmidt, R. Danescu, D. Frentiu, T. Marita, T. Graf, F. Oniga, and C. Pocol. High Accuracy Stereo Vision System for Far Distance Obstacle Detection. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 292–297, 2004b. (Cited on page 32.)

D. Nehab, S. Rusinkiewiez, and J. Davis. Improved Sub-Pixel Stereo Correspondences Through Symmetric Refinement. *International Conference on Computer Vision (ICCV)*, pages 557–563, 2005. (Cited on pages 28 and 85.)

J. Neyman and E. Pearson. On the Problem of the Most Efficient Tests of Statistical Hypotheses. *Mathematical, Physical and Engineering Sciences*, 231, 1933. (Cited on page 16.)

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999. (Cited on pages 7, 11, and 12.)

F. Oniga and S. Nedevschi. Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle and Obstacle Detection. *Vehicular Technology*, 2010. (Cited on page 32.)

D. Pfeiffer and U. Franke. Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data. In *British Machine Vision Conference (BMVC)*, 2011. (Cited on pages viii, 32, 34, 35, 57, 68, 74, 76, 118, and 121.)

D. Pfeiffer, S. Gehrig, and N. Schneider. Exploiting the Power of Stereo Confidences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 297–304, 2013. (Cited on page 85.)

P. Pinggera, U. Franke, and R. Mester. Highly Accurate Depth Estimation for Objects at Large Distances. In *German Conference on Pattern Recognition (GCPR)*, pages 21–30, 2013. (Cited on pages 83, 86, and 90.)

P. Pinggera, D. Pfeiffer, U. Franke, and R. Mester. Know Your Limits: Accuracy of Long Range Stereoscopic Object Measurements in Practice. In *European Conference on Computer Vision (ECCV)*, pages 96–111, 2014. (Cited on pages 83, 98, and 105.)

P. Pinggera, U. Franke, and R. Mester. High-Performance Long Range Obstacle Detection Using Stereo Vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015. (Cited on pages 31, 65, and 67.)

P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester. Lost and Found: Detecting Small Road Hazards for Self-Driving Vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016. (Cited on pages 31, 57, 64, 65, and 66.)

G. W. Pulford. Taxonomy of Multiple Target Tracking Methods. *IEE Proceedings Radar, Sonar, and Navigation*, 152(5):291–304, 2005. (Cited on page 61.)

C. Rabe. *Detection of Moving Objects by Spatio-Temporal Motion Analysis*. Phd thesis, Christian-Albrechts-Universität zu Kiel, 2011. (Cited on pages 24, 30, 61, 98, 102, and 115.)

S. Ramos, S. Gehrig, P. Pinggera, U. Franke, and C. Rother. Detecting Unexpected Obstacles for Self-Driving Cars: Fusing Deep Learning and Geometric Modeling. In *IEEE Intelligent Vehicles Symposium (IV)*, 2017. (Cited on pages 64, 65, 82, and 119.)

R. Ranftl, S. Gehrig, T. Pock, and H. Bischof. Pushing the Limits of Stereo Using Variational Stereo Estimation. In *IEEE Intelligent Vehicles Symposium (IV)*, 2012. (Cited on pages 85 and 101.)

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 185–192, 2015. (Cited on page 33.)

D. Robinson and P. Milanfar. Fundamental Performance Limits in Image Registration. *IEEE Transactions on Image Processing (TIP)*, 13(9):1185–1199, 2004. (Cited on pages 85, 86, and 89.)

J. Rodriguez and J. Aggarwal. Stochastic Analysis of Stereo Quantization Error. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 12(May), 1990. (Cited on page 28.)

P. J. Rousseeuw and C. Croux. Alternatives to the Median Absolute Deviation. *Journal of the American Statistical Association*, 88(424), 1993. (Cited on page 104.)

N. Sabater, J.-M. Morel, and A. Almansa. How Accurate Can Block Matches Be in Stereo Vision? *SIAM Journal on Imaging Sciences*, 4(1): 472, 2011. (Cited on pages 85 and 108.)

N. Sabater, A. Almansa, and J.-M. Morel. Meaningful Matches in Stereovision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(5):930–42, may 2012. (Cited on page 85.)

SAE. J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, 2016. URL `https://www.sae.org`. (Cited on page 2.)

H. S. Sawhney. 3D Geometry from Planar Parallax. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994. (Cited on page 32.)

H. Scharr. Optimal Filters for Extended Optical Flow. In *LNCS 3417 - Complex Motion*, volume 1114, pages 14–29. Springer, 2007. (Cited on pages 85 and 89.)

D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision (IJCV)*, 47(1-3):7–42, 2002. ISSN 0920-5691. (Cited on pages 25, 28, 83, and 84.)

D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth. In *German Conference on Pattern Recognition (GCPR)*, 2014. (Cited on pages 25, 26, 28, and 84.)

T. Scharwächter and U. Franke. Low-Level Fusion of Color, Texture and Depth for Robust Road Scene Understanding. In *IEEE Intelligent Vehicles Symposium (IV)*, 2015. (Cited on page 33.)

L. Schneider, M. Cordts, T. Rehfeld, D. Pfeiffer, M. Enzweiler, U. Franke, M. Pollefeys, and S. Roth. Semantic Stixels: Depth is Not Enough. In *IEEE Intelligent Vehicles Symposium (IV)*, 2016. (Cited on pages viii, ix, 33, 118, and 119.)

T. Schoenemann and D. Cremers. Near Real-Time Motion Segmentation Using Graph Cuts. In *DAGM Symposium*, 2006. (Cited on page 91.)

A. Seki and M. Pollefeys. SGM-Nets: Semi-Global Matching with Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on page 26.)

M. Shimizu and M. Okutomi. Precise Sub-pixel Estimation on Area-Based Matching. In *International Conference on Computer Vision (ICCV)*, pages 90–97, 2001. (Cited on pages 28, 85, and 103.)

G. Sibley, L. Matthies, and G. Sukhatme. Bias Reduction and Filter Convergence for Long Range Stereo. In S. Thrun, R. Brooks, and H. Durrant-Whyte, editors, *Robotics Research*, pages 285–294. Springer Berlin Heidelberg, 2007. (Cited on pages 29 and 30.)

S. Sivaraman and M. M. Trivedi. Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis. *IEEE Transactions on Intelligent Transportation Systems (TITS)*, 14(4): 1773–1795, 2013. (Cited on page 33.)

Statistisches Bundesamt. Verkehrsunfälle 2016. Technical report, 2016. URL `https://www.destatis.de`. (Cited on page 1.)

Z. Sun, G. Bebis, and R. Miller. On-Road Vehicle Detection: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(5):694–711, 2006. (Cited on page 33.)

M. A. Sutton, J.-J. Orteu, and H. W. Schreier. *Image Correlation for Shape, Motion and Deformation Measurements*. Springer, 2009. (Cited on pages 51, 85, 86, 87, and 111.)

R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010. (Cited on page 26.)

R. Szeliski and D. Scharstein. Sampling the Disparity Space Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(3): 419–25, 2004. (Cited on page 84.)

P. Thévenaz, T. Blu, and M. Unser. Interpolation Revisited. *IEEE Transactions on Medical Imaging (TMI)*, 19(7), 2000. (Cited on pages 85 and 89.)

S. Thrun. Stanley: The Robot that Won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9), 2006. (Cited on page 2.)

C. Tomasi and T. Kanade. Detection and Tracking of Point Features. Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991. (Cited on pages 45, 64, and 86.)

B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle Adjustment - A Modern Synthesis. *Vision Algorithms: Theory and Practice (LNCS)*, 1883:298–372, 2000. (Cited on pages 38 and 41.)

E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998. (Cited on pages 24 and 27.)

R. Y. Tsai. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987. (Cited on page 23.)

M. Unser, A. Aldroubi, and M. Eden. B-Spline Signal Processing. *IEEE Transactions on Signal Processing (TSP)*, 41(2), 1993. (Cited on page 89.)

C. Urmson. Autonomous Driving in Urban Environments: Boss and the Urban Challenge. *Journal of Field Robotics*, 25(8), 2008. (Cited on page 2.)

C. Vogel, K. Schindler, and S. Roth. 3D Scene Flow Estimation with a Piecewise Rigid Scene Model. *International Journal of Computer Vision (IJCV)*, 115(1), 2015. (Cited on pages 26 and 85.)

A. Wedel, H. Badino, C. Rabe, H. Loose, U. Franke, and D. Cremers. B-Spline Modeling of Road Surfaces with an Application to Free Space Estimation. *IEEE Transactions on Intelligent Transportation Systems (TITS)*, 2009a. (Cited on pages 32 and 35.)

A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. An Improved Algorithm for TV-L1 Optical Flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23–45. Springer, 2009b. (Cited on pages 91, 96, 101, 102, and 123.)

M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 Optical Flow. In *British Machine Vision Conference (BMVC)*, pages 108.1–108.11, 2009. (Cited on pages 101 and 102.)

T. Williamson and C. Thorpe. A Trinocular Stereo System for Highway Obstacle Detection. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1999. (Cited on pages 32 and 46.)

K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation. In *European Conference on Computer Vision (ECCV)*, pages 756–771, 2014. (Cited on page 26.)

S. Yang, B. Bhanu, and A. I. Mourikis. Error Model for Scene Reconstruction from Motion and Stereo. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 70–77. Ieee, 2010. (Cited on page 29.)

J. Žbontar and Y. LeCun. Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. *Journal of Machine Learning Research (JMLR)*, 2016. (Cited on page 25.)

T. Zhang and T. Boult. Realistic Stereo Error Models and Finite Optimal Stereo Baselines. In *IEEE Workshop on Applications of Computer Vision*, 2011. (Cited on page 29.)

Z. Zhang. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(11): 1330–1334, 2000. (Cited on pages 22 and 23.)

Z. Zhang, R. Weiss, and A. Hanson. Obstacle Detection Based on Qualitative and Quantitative 3D Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(1):15–26, 1997. (Cited on page 32.)

W. Zhao and N. Nandhakumar. Effects of Camera Alignment Errors on Stereoscopic Depth Estimates. *Pattern Recognition*, 29(12), 1996. (Cited on page 27.)

J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, E. Kaus, R. G. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knoppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb. Making Bertha Drive - An Autonomous Journey on a Historic Route. *IEEE Intelligent Transportation Systems Magazine*, 6(2):8–20, jan 2014. (Cited on page 2.)