

**Einsatz der PMD-Kamera in der mobilen Robotik
für die Hinderniserkennung und -vermeidung sowie
für die Selbstlokalisierung**

**Der Naturwissenschaftlich-Technischen Fakultät der
Universität Siegen**

zur Erlangung des akademischen Grades

**Doktor der Ingenieurwissenschaften
(Dr.-Ing.)**

genehmigte Dissertation

von

Dipl.-Ing. Alexander Prusak

1. Gutachter: Univ.-Prof. Dr.-Ing. Hubert Roth

2. Gutachter: Univ.-Prof. Dr.-Ing. Günter Schröder

Vorsitzender der Prüfungskommission: Univ.-Prof. Dr.-Ing. Robert Mayr

Tag der mündlichen Prüfung: 21. November 2012

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Regelungs- und Steuerungstechnik an der Universität Siegen unter der Betreuung von Prof. Dr.-Ing. Hubert Roth.

Herrn Prof. Dr.-Ing. Hubert Roth danke ich für das Zustandekommen dieser Arbeit und insbesondere für die Möglichkeit, auf dem Gebiet der mobilen Robotik in mehreren Forschungsprojekten tätig zu sein.

Meine besondere Hochachtung gilt Herrn Prof. Dr.-Ing. Rudolf Schwarte, der mit seiner Erfindung der PMD (Photonic Mixer Device)-Kamera meiner Arbeit die Basis gegeben hat. Auch bin ich ihm für die zahlreichen und umfangreichen Diskussionen und Anregungen dankbar, die auch diese Niederschrift beeinflusst haben.

Den Studenten der Elektrotechnik, der Informatik und auch der Mechatronik danke ich, die mich mit ihren Diplom- und Studienarbeiten unterstützt haben. Die gute Zusammenarbeit mit den Kollegen des Instituts sowie die Kooperation mit der Fa. Götting KG möchte ich ebenfalls herausstellen.

Und schließlich danke ich meiner Partnerin und unserer Tochter Henrike E. Groos für die Geduld und Anteilnahme, die sie in den letzten Monaten hierfür aufgebracht haben.

Siegen, im November 2012

Alexander Prusak¹

¹ www.prusak.de - alexander@prusak.de

Inhalt

Abstract	ix
Formelzeichen und Konstanten	xi
Verwendete Abkürzungen und Begriffe	xvi
1 Einleitung	1
1.1 Überblick	1
1.2 Problemstellung der Arbeit	5
1.3 Motivation	7
1.4 Aufbau der Arbeit	8
1.5 Wissenschaftlicher Beitrag	8
2 Die PMD-Kamera	10
2.1 Varianten der PMD-Kamera und andere 3D-Kameras	11
2.2 Kalibrierung der PMD-Kamera	13
2.3 Funktionsweise der PMD-Kamera	15
2.3.1 Integrationszeit	18
2.3.2 Eindeutigkeitsbereich	20
2.4 Vorverarbeitung und Filterung von PMD-Videobilder	21
2.5 2D- und 3D-Videobilder und deren Kombination	21
3 Hindernis- und Fahrwegerkennung mit der PMD-Kamera	25
3.1 Integration der PMD-Kamera in ein mobiles Fahrzeug	25
3.2 Ausnahmesituationen	27
3.3 Varianten von Fahrwegen	29
3.4 Hindernis- und Fahrwegerkennung	32
4 Die mobilen Roboter Tom3D und Merlin3D	36
4.1 Fahrantriebe	36
4.2 Konventionelle und bildgebende Sensorik	38
4.2.1 Mechanischer Front-Bumper	38
4.2.2 IR-Front- und IR-Boden-Bumper	39
4.2.3 Inklinationssensor	40

4.2.4 Rad-Encoder	41
4.2.5 2D-Kamera	41
4.2.6 Omnidirektionale Kamera	42
4.2.7 Stereovision	43
4.3 Hard- und Software-Architektur	43
4.3.1 Motor-Control-Board	45
4.3.2 Geschwindigkeitsregelung	45
4.3.3 Sensor-Server	47
4.3.4 Video-Server	48
4.3.5 Vernetzung, Telemetrie und Kommunikation	49
4.3.5.a Topologie eines vernetzten mobilen Fahrzeugs	49
4.3.5.b Teleaktivierung und -deaktivierung des mobilen Fahrzeugs	53
4.3.5.c Datenverbindungen via WWW-, FTP-, Telnet- und Remotedesktop	55
4.3.5.d Bewegungssteuerung	56
4.3.5.e Graphical User Interface (GUI)	59
4.3.6 Roboter-Controller	61
4.4 Sicherheitsaspekte	63
5 Lokale Navigation	67
5.1 Hardware-nahe Hinderniserkennung	67
5.2 Hindernisvermeidung mit Fuzzy-Logic-Regelung	68
5.2.1 Fuzzyfizierung	71
5.2.2 Fuzzy-Inferenz: Regelbasis	73
5.2.3 Fuzzy-Inferenz: Auswertung der Regelprämisse	75
5.2.4 Fuzzy-Inferenz: Regelaktivierung	76
5.2.5 Fuzzy-Inferenz: Aggregation	77
5.2.6 Defuzzyfizierung	77
5.2.7 Konvertierung der Ausgangsgrößen bzw. Stellgrößen	78
6 Selbstlokalisierung	81
6.1 Koordinatensysteme	82
6.2 Pose eines mobilen Fahrzeugs	84

6.3 Absolute Selbstlokalisierung	84
6.4 Relative Selbstlokalisierung	88
6.5 Odometrie mit Rad-Encoder	89
6.6 Odometrie mit einer 2D-Kamera (visuelle 2D-Odometrie)	91
6.7 Kumulierung des Fehlers bei der Koppelnavigation	93
6.8 Selbstlokalisierung mit der PMD-Kamera	96
6.8.1 Merkmalsextraktion	98
6.8.2 Korrespondenzbildung	101
6.8.3 Bestimmung der Transformation	104
6.8.3.a Bestimmung der Rotation mittels Einheitsquaternionen-Methode	106
6.8.3.b Bestimmung der Rotation mittels HAYAI-Matching-Methode	108
6.8.3.c Bestimmung der Translation	109
6.8.4 ICP (Iterative Closest Points) -Methode	109
6.8.5 Diskussion	111
7 Navigation und Dienstleistung	114
7.1 Lokale Navigation	115
7.1.1 Fahrmanöver	115
7.1.2 Wand folgen	116
7.1.3 Geometrische Figuren fahren	116
7.1.4 Hinderniserkennung und -vermeidung	117
7.1.5 Selbstlokalisierung	117
7.2 Globale Navigation und Dienstleistungen	118
7.2.1 Teilautonomie	118
7.2.2 Treppen hinauf- und herabfahren	119
7.2.3 Pose einnehmen	120
7.2.4 Zufallsfahren	121
7.2.5 Exploration und Kartengenerierung	121
7.2.6 Pfad- bzw. Bahnplanung und Pfad- bzw. Bahnfahrt ohne/mit Karte	123
7.2.7 Andocken ohne/mit Karte	124
7.2.8 Vollautonomie	125

8 Ausblick	126
8.1 Systematik (teil)autonomer Funktionen und Dienstleistungen	126
8.2 Absolute Selbstlokalisierung mit Hilfe des WLAN	128
9 Zusammenfassung	130
10 Anhang	133
10.1 Matrizen	133
10.1.1 Rotationsmatrizen	133
10.1.2 Homogene Rotationsmatrizen	133
10.2 Multivariate Normalverteilung	133
10.2.1 Empirischer arithmetischer Mittelwert	133
10.2.2 Varianz und Kovarianz zweier Zufallsvariablen X und Y	134
10.2.3 Empirische Varianz und Kovarianz zweier Zufallsvariablen X und Y	134
10.2.4 Kovarianzmatrix zweier Zufallsvariablen X und Y	134
10.2.5 Korrelationskoeffizient zweier Zufallsvariablen X und Y	134
10.3 Transformation einer multivariaten Normalverteilung	134
10.3.1 Lineare Transformation	134
10.3.2 Nichtlineare Transformation	135
10.3.3 Funktion der Pose bzw. der Poseänderung	136
10.3.4 Kovarianzmatrix Σ_{v+1}	136
10.4 Quaternionen	137
11 Literaturverzeichnis und Verweise	141

Abstract

The subject of this thesis is the obstacle and driveway recognition, the obstacle avoidance and the self-localization of a mobile vehicle. In the past therefore were used 2D, 3D laser scanner and stereo vision. This thesis shows the application of the PMD camera.

As experimental platform was presented a mobile vehicle with differential drive (Tom3D) and another vehicle with Ackermann-drive (Merlin3D). Both mobile vehicles are equipped with the same system architecture, and each has an embedded PC for sensor data processing, image processing and communications, as well as a C167 microcontroller for speed control and obstacle avoidance. For the speed control and for a rough estimate of self-localization there are used wheel encoders. GPS and inclination sensors provide the self-localization. The access to such a mobile vehicle by the operator is possible as a manual control or can be realized by Wireless-LAN (WLAN). A tele-presence respectively telemetry is conceivable via mobile radio or satellite radio.

A PMD camera and a 2D camera were integrated with a defined inclination angle onto the mobile vehicle. The unambiguity interval of 7.5 m, the optical field of view and the mounting height of the PMD-camera were considered. From this perspective, the PMD camera always records obstacles and driveway together in the front area of the mobile vehicle. There must be a differentiation between obstacles and the driveway within the PMD-video image. It also applies to differentiate between so-called negative and positive obstacles. For a real-time separation of driveway, negative and positive obstacles at every PMD video image must consider the inclination angle of the PMD-camera, as well as pitch and roll angle of the mobile vehicle.

After successful extraction of the driveway and negative and positive obstacles inside the PMD-video image, it is divided into equal vertical segments. The segment (segment number) with the smallest distance will be selected.

The segment number and the minimum distance value are the linguistic input variables for the Fuzzy-Logic-Controller. After the fuzzification of these values follows the fuzzy-inference, in this are the driving manoeuvres linguistically formulated for obstacle avoidance. Finally within the defuzzification the result data are converted into concrete motor control signals. With the help of these steps navigation is realizable without environment map and without path or driveway control. The mobile vehicle thereby can be already used for uncontrolled driving or random driving for tasks of exploration.

Only with the availability of a self-localization, here specifically the relative self-localization, path driving and map generation are possible. Movements of a mobile vehicle can be observed by position changes of objects or their significant features within the continuous PMD image sequence of a PMD camera. These position changes must be determined within two sequential PMD video images. The problem is, however, to find the significant features of the first PMD video image again in the second PMD video image and assign them to each other. From these position changes the translation vector and rotation matrix respectively the pose change (position and orientation) of the mobile vehicle can be calculated.

Significant features within the participating PMD-video images can be extracted with the help of the so-called Moravec- respectively Interest-operator. Now the correspondence between two feature point sets can be done by using the Euclidean distance respectively Euclidean norm. Based on these pairings, and by using the Unit-Quaternion-Method for 3D movements or by using the HAYAI-Method for 2D movements can be determined the desired rotation matrix and thereafter the translation vector. Translation vector and rotation matrix still have errors, which can be reduced iteratively by using the ICP-Method.

Simple manoeuvres, obstacle detection, obstacle avoidance, self-localization, path driving, etc. can be combined with each other and thus create semi-autonomous functions. A hierarchical compilation will show the transition from pure remote control, via semi-autonomous and autonomous functions to useful services with the help of a mobile vehicle.

Formelzeichen und Konstanten

\varnothing, d_{Rad}	Durchmesser
$\alpha, \alpha_1, \alpha_2, \alpha_i, \alpha_j$ $\alpha_{LF}, \alpha_{LLL}, \alpha_{LRL}$ $\alpha_{NF}, \alpha_{NULL}, \alpha_{RF}, \alpha_{SF}$ $\alpha_{STOP}, \alpha_{WLL}, \alpha_{WRL}$	Erfüllungs- bzw. Aktivierungsgrade des Fuzzy-Logic-Reglers
$\alpha, \alpha_F, \alpha_{F_Obj}, \alpha_v$ $\alpha_{Obj}, \beta, \beta_F, \beta_{F_Obj}$ $\beta_{Obj}, \gamma, \gamma_F, \gamma_{F_Obj}$ $\gamma_{Obj}, \Phi_F, \Phi_{F_Obj}, \Phi_{Obj}$	Orientierungswinkel des mobilen Fahrzeugs/eines Objekts
$\Delta\alpha, \Delta\beta, \Delta\gamma$ $\Delta\alpha_v, \Delta\beta_v, \Delta\gamma_v$	Änderung des Orientierungswinkels des mobilen Fahrzeugs
a	Amplitude der Kreuzkorrelationsfunktion
a_{Rad}	halber Radabstand beim Differentialantrieb
A	Matrix von Konstanten eines Übertragungsfunktionssystems
A_1, A_2, A_3, A_4	Abtastwerte der Kreuzkorrelationsfunktion
A_{1i}, A_{2i}	Teilschwerpunkte der resultierenden Zugehörigkeitsfunktionen $\mu_{1maxmin}$ und $\mu_{2maxmin}$
b	Vektor von Konstanten eines Übertragungsfunktionssystems
b	Offset der Kreuzkorrelationsfunktion
c	Lichtgeschwindigkeit ($c = 2,9979 \cdot 10^8$ m/s \approx 300.000 km/s)
c_1, c_2, k	Kalibrierungskonstanten
$\mathbf{c}_d, c_{dx}, c_{dy}, c_{dz}$ $\mathbf{c}_m, c_{mx}, c_{my}, c_{mz}$	Bildschwerpunkte als Vektor und als x -, y -, z -Komponenten
$d, d_{min}, d_{minF}, d_{min0}$ d_{max}	allgemeiner, minimaler, fuzzyfizzierter, bestimmter und maximaler Distanzwert
d_0	maximale Frontausdehnung des mobilen Fahrzeugs
d_h, d_v	horizontale und vertikale Abmessung des PMD-Chip
d_K	Montageabstand der PMD-Kamera
D, M, M'	normale u. verkleinerte 2D-Grauwertbilder ² („Daten“ und „Muster“) der 2D-Odometrie
D, D_{neu}, M	alle markante Merkmale zweier aufeinander folgenden PMD-Videobilder („Daten“ und „Muster“)
$d(x,y), d_{2D}(x,y)$ $\mathbf{d}(x,y,z), \mathbf{d}_{3D}(x,y,z)$	PMD-Pixel eines PMD-Videobildes („Daten“)
$\mathbf{d}^*(x,y,z), \mathbf{d}^*, \mathbf{d}^{**}$	PMD-Pixel eines PMD-Videobildes („Daten“) nach einer bzw. nach zwei Rotationsoperationen

² **M'** stellt ein verkleinertes 2D-Subbild mit $2k'+1 \times 2l'+1$ des 2D-Videobildes **M** mit $2k+1 \times 2l+1$ (Höhe \times Breite) dar.

$\mathbf{d}_i, d'_{xi}, d'_{yi}, d'_{zi}$ $\mathbf{d}'_i, \mathbf{d}_{ineu}$	i -tes markantes Merkmal ³ innerhalb eines PMD-Videobildes
$\dot{\mathbf{d}}'_i, \dot{\mathbf{d}}'_i, \dot{\mathbf{D}}'_i, \dot{\mathbf{D}}'_i$	i -tes markantes Merkmal ³ als Quaternion in Vektor-, Matrixform und deren Konjugation
ε	Variationsbreite bei der 2D-Kreuzkorrelation der 2D-Odometrie
$E(\mathbf{x}), E(\mathbf{y})$	Erwartungswerte
$\mathbf{e}, \mathbf{e}_1, \mathbf{e}_2, \mathbf{E}$	Einheitsvektoren, Einheitsmatrix
$\mathbf{e}_i, E, E_{alt}, \Delta E, E_{neu}$	i -ter Fehlervektor, Fehlerfunktionen
$\hat{\mathbf{e}}, \hat{\mathbf{e}}_{max}$	Eigenvektoren als Quaternionen
f	optischer Fokus
f_m, T_m	Modulationsfrequenz und -phasendauer der PMD-Kamera
$\mathbf{f}(\mathbf{x}), \mathbf{f}(\mu_x)$	Übertragungsfunktionssystem
$\mathbf{f}(\mu_v), \mathbf{f}(\mu_{\Delta v}), \mathbf{f}(\mathbf{p}_{v+1})$	Funktionen der Pose bzw. der Poseänderung
$g(\mathbf{p}_v), g(\Delta \mathbf{p}_v)$ $g(\mathbf{p}_{v+1})$	Multivariate Normalverteilungen der Pose und Poseänderung
h_K	Montagehöhe der PMD-Kamera
i, j, k, l, m, n, v	Indices allgemein
$\mathbf{i}, \mathbf{j}, \mathbf{k}$	hyperkomplexe Elemente einer Quaternion
j_{min}	Index bei der Korrespondenzbildung mittels Euklidischer Distanz
$i_{center}, i_{kk_{max}}, j_{center}$ $j_{kk_{max}}, k', l'$	Indices bei der 2D-Kreuzkorrelation der 2D-Odometrie ²
I_l, I_r, I_{max}	Rad-Encoder-Impulse
K_D, K_I, K_P, K_R, K_S	Differenzier-, Integrier-, Proportional-, Regler-, Streckenverstärkung
kk, kk', kk_{max}, kkf	2D-Kreuzkorrelation ² , Kreuzkorrelationsfunktion
$Kov(X,X), Kov_n(X,X)$ $Kov(X,Y), Kov_n(X,Y)$ $Kov(Y,X), Kov_n(Y,X)$ $Kov(Y,Y), Kov_n(Y,Y)$	Kovarianz zweier Zufallsvariablen X und Y
l_{PMD}	Eindeutigkeitsbereich der PMD-Kamera
l^2, l_1, l_2	Mahalanobis-Distanz
l_i, l_{ij}, l_{ijmin}	Euklidische Distanz
$\lambda, \lambda_1, \lambda_2, \lambda_3, \lambda_4$ λ_i, λ_{max}	Eigenwertvektor, Eigenwerte und größter Eigenwert
$m(x,y)$	PMD-Pixel eines PMD-Videobildes („Muster“)
$\mathbf{m}'_i, m'_{xi}, m'_{yi}, m'_{zi}$	i -tes markantes Merkmal ³ eines PMD-Videobildes („Muster“)

³ Aus den markanten Merkmalen zweier aufeinander folgender PMD-Videobilder \mathbf{d}_i und \mathbf{m}_j werden durch Überlagerung der jeweiligen Bildschwerpunkte \mathbf{c}_d und \mathbf{c}_m die neuen Punktmengen \mathbf{d}'_i und \mathbf{m}'_j gebildet.

$\mathbf{m}_j, \mathbf{m}^{2Dj}, \mathbf{m}^{x2Dj}, \mathbf{m}^{y2Dj}$ j -tes markantes Merkmal³ eines PMD-Videobildes („Muster“)
 $\mathbf{m}^{3Dj}, \mathbf{m}'_j, \mathbf{m}'_{jmin}$

$\dot{\mathbf{m}}'_i, \ddot{\mathbf{m}}'_i, \dot{\mathbf{M}}'_i, \ddot{\mathbf{M}}'_i$ i -tes markantes Merkmal³ als Quaternion in Vektor-, Matrixform und deren Konjugation

$\mu_i, \mu_j, \mu_\Delta, \mu_\square, \mu_{FD}$ Zugehörigkeitsfunktionen des Fuzzy-Logic-Reglers
 $\mu_{GD}, \mu_{LF}, \mu_{LLL}, \mu_{LRL}$
 $\mu_{MD}, \mu_{ML}, \mu_{MR}, \mu_{ND}$
 $\mu_{NF}, \mu_{NULL}, \mu_{RF}, \mu_{SF}$
 $\mu_{STOP}, \mu_{UD}, \mu_{WL}$
 $\mu_{WLL}, \mu_{WR}, \mu_{WRL}$

$\mu_{imaxmin}, \mu_{1maxmin}$ Aggregation der Zugehörigkeitsfunktionen mit Hilfe der
 $\mu_{jmaxmin}, \mu_{2maxmin}$ MAX-MIN-Inferenz-Methode

μ_{imin}, μ_{1min} Aktivierung der Zugehörigkeitsfunktionen mit Hilfe der
 μ_{jmin}, μ_{2min} MIN-Methode nach Mamdani

$\mu_{LFmin}, \mu_{LLLmin}$ Aktivierung der Zugehörigkeitsfunktionen mit Hilfe der
 $\mu_{LRLmin}, \mu_{NFmin}$ MIN-Methode
 $\mu_{NULLmin}, \mu_{RFmin}$
 $\mu_{SFmin}, \mu_{STOPmin}$
 $\mu_{WLLmin}, \mu_{WRLmin}$

μ_{iprod}, μ_{jprod} Aktivierung der Zugehörigkeitsfunktionen mit Hilfe der
 PROD-Methode

$\boldsymbol{\mu}_v, \boldsymbol{\mu}_{v+1}, \boldsymbol{\mu}_{\Delta v}, \boldsymbol{\mu}_x, \boldsymbol{\mu}_y$ Erwartungswertvektoren einer multivariaten Normalverteilung

$\mu_{xv}, \mu_{yv}, \mu_{\alpha v}$ Erwartungswerte einer multivariaten Normalverteilung
 $\mu_{\Delta \alpha v}, \mu_{\Delta s v}$

$\mathbf{n}_F, \mathbf{n}_{F0}, \mathbf{n}_{hF0}$ (homogener) Normalenvektor des mobilen Fahrzeugs
 \mathbf{N} Kreuzkovarianzmatrix

N, N_d, N_m Anzahl der markanten Merkmale

$N_{Segment}, N_{Segment0}$ allgemeine, bestimmte und fuzzyfizierte Segment-Nummer
 $N_{SegmentF}$

$\nabla_{\Delta v}, \nabla_v, \nabla_x$ Nablaoperatoren

$\Delta\varphi, \Delta\varphi_{max}$ Phasendifferenz

φ_{Fe2} Winkel zwischen Fahrzeugvektor \mathbf{n}_F und Einheitvektor \mathbf{e}_2

$\varphi_h, \varphi_v, \varphi_{Opt.}$ horizontaler und vertikaler optische Öffnungswinkel

$\varphi_{Pitch}, \varphi_{Roll}, \varphi_{Yaw}$ Pitch-, Roll- und Yaw-Winkel bzw. Nick-, Roll- und Gierwinkel

φ_{PMD} Neigungswinkel der PMD-Kamera

$\Delta\mathbf{p}_v, \Delta\mathbf{p}_{3Dv}, \Delta\mathbf{p}_{6Dv}$ v -te Poseänderung, v -te 3D-, 6D-Poseänderung

$\mathbf{p}_0, \mathbf{p}_v, \mathbf{p}_{v+1}$ Anfangspose, v -te, $v+1$ -te Pose bzw. 3D-, 6D-Pose

$\mathbf{p}_{3D}, \mathbf{p}_{6D}$

$\mathbf{p}_{3Dn}, \mathbf{p}_{6Dn}$ 3D- bzw. 6D-Pose nach n Poseänderungen

P_{opt} Lichtsignalleistung

PWM_l, PWM_r PWM_φ^*	PWM-Signale für den linken und den rechten Antriebsmotor beim Differentialantrieb
PWM_v, PWM_φ	PWM-Signale für den Antriebsmotor und den Lenk-Servoantrieb beim Ackermannantrieb
p, q	Höhe und Breite der Filtermaske des Moravec-Operator
$\dot{\mathbf{q}}, \dot{\mathbf{Q}}, \ddot{\mathbf{q}}, \ddot{\mathbf{Q}}$	Quaternion in Vektor-, Matrixform und deren Konjugation
$\rho(X, Y), \rho_n(X, Y)$	Korrelationskoeffizient zweier Zufallsvariablen X und Y
\mathbf{r}	Vektor zwischen den Bildschwerpunkten \mathbf{c}_d und \mathbf{c}_m
$\mathbf{r}_F, \mathbf{r}_{hF}$	Normalenvektor und homogener Normalenvektor des mobilen Fahrzeugs im Weltkoordinatensystem
r_{Rad}	Radradius
$\mathbf{R}_h, \mathbf{R}_{hx}, \mathbf{R}_{hy}, \mathbf{R}_{hz}$	homogene Rotationsmatrizen
$\mathbf{R}, \mathbf{R}_{Chip}, \mathbf{R}_F, \mathbf{R}_{Fn}$	Rotationsmatrizen
$\mathbf{R}_K, \mathbf{R}_{K_Ref}, \mathbf{R}_{Ref}, \mathbf{R}_v$	
$\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z$	
$\mathbf{R}_{Si}, \mathbf{R}_{zi}, \mathbf{R}_{yi}, \mathbf{R}_{xi}$	Rotationsmatrizen des i -ten Sensors
$\Delta s_l, \Delta s_r, \Delta s_v, \Delta s_{v+1}$	Linkes, rechtes, v -tes und $v+1$ -tes Streckenstück
$\Sigma_{\Delta v}, \Sigma_v, \Sigma_{v+1}, \Sigma_x$ Σ_y	Kovarianzmatrix einer multivariaten Normalverteilung
$\sigma_{xx}, \sigma_x^2, \sigma_{xy}$ $\sigma_{yx}, \sigma_{yy}, \sigma_y^2$ $\sigma_{xv}^2, \sigma_{xyv}, \sigma_{xav}, \sigma_{yxv}$ $\sigma_{yv}^2, \sigma_{yav}, \sigma_{axv}, \sigma_{ayv}$ $\sigma_{av}^2, \sigma_{\Delta s_v}^2, \sigma_{\Delta s \Delta a_v}$ $\sigma_{\Delta a \Delta s_v}, \sigma_{\Delta a_v}^2$	Varianz- und Kovarianzelemente der Kovarianzmatrix
s, s_{error}	Schwellwerte
$S_{xx}, S_{xy}, S_{xz}, S_{yx}$ $S_{yy}, S_{yz}, S_{zx}, S_{zy}, S_{zz}$	Elemente der Kreuzkovarianzmatrix N
t	Zeit
$\mathbf{t}, \mathbf{t}_{Chip}, \mathbf{t}_F, \mathbf{t}_{F_Obj}$ $\mathbf{t}_{Fn}, \mathbf{t}_K, \mathbf{t}_{K_Ref}, \mathbf{t}_{Obj}$ $\mathbf{t}_{Ref}, \mathbf{t}_{Si}, \mathbf{t}_v$	Translationsvektoren
$\mathbf{T}, \mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_{Chip}, \mathbf{T}_F$ $\mathbf{T}_{Fn}, \mathbf{T}_K, \mathbf{T}_{K_Ref}, \mathbf{T}_n$ $\mathbf{T}_{n-1}, \mathbf{T}_{Ref}, \mathbf{T}_{Si}, \mathbf{T}_v$	Transformationsmatrizen
$T_{abstast}, T_N, T_S, T_t, T_V$	Abtast-, Nachstell-, Streckenverzögerungs-, Tot-, Vorhaltezeit
${}^{(a)}t_i, {}^{(b)}t_i, \dots, {}^{(g)}t_i, t_{ICP}$	Zeitdauer der Prozessschritte (a) bis (g) des ICP-Algorithmus,
t_{Frame}	Frame-Taktzeit der PMD-Kamera
t_{TOF}, τ	Echolaufzeit, Time-of-Flight (TOF)
T_{int}	Integrationszeit der PMD-Kamera

\mathbf{u}	Einheitsvektor einer Quaternion
$U_0, U_a, U_b, U_\Delta, U_\Sigma$	Spannungswerte beim Mischprozess in einem PMD-Pixel
U_{in}	Eingangsspannung
u_m, u_{ma}, u_{mb}	Modulationssignale innerhalb eines PMD-Pixels
U_{Rad}	Umfang eines Rades
\ddot{u}	Getriebeübersetzung
$V, V_0, V_{45}, V_{90}, V_{135}$	quadratische Gradientensummen des Moravec-Operator
v_F	Fahrgeschwindigkeit des mobilen Fahrzeugs
$\text{Var}(X), \text{Var}_n(X)$ $\text{Var}(Y), \text{Var}_n(Y)$	Varianz der Zufallsvariablen X und Y
w	Führungsgröße (Sollwert) eines Regelkreises
\mathbf{x}, \mathbf{y}	Eingangs- und Ausgangsvektor eines Übertragungsfunktionssystems
\bar{x}_n, \bar{y}_n	empirischer arithmetischer Mittelwert
X, Y, x_i, y_i	Zufallsvariablen und ihre möglichen Werte ($i = 1, \dots, n$)
$x_{2D}, y_{2D}, z(x_{2D}, y_{2D})$	PMD-Pixel im 2D-Bildspeicher
x_{3D}, y_{3D}, z_{3D}	PMD-Pixel im 3D-Bildspeicher
x_{Chip}, y_{Chip} $z(x_{Chip}, y_{Chip})$	PMD-Pixel innerhalb des PMD-Chip
$\Delta x_v, \Delta y_v, \Delta z_v$	v -te Positionsänderung
$x_d, x_1, x_2, x_3, x_4, x_5$	Regeldifferenz, Regelgrößen 1 - 5
$x_{\Delta a}, x_{\Delta m}, x_{\Delta e}$ $y_{\Delta a}, y_{\Delta m}, y_{\Delta e}$	Anfangs-, Mittel- und Endwert einer dreiecksförmigen Zugehörigkeitsfunktion μ_Δ
x_K, y_K, z_K	Kamerakordinaten im Fahrzeugkoordinatensystem
x_{Fi}, y_{Fi}, z_{Fi} $x_{F_Obj}, y_{F_Obj}, z_{F_Obj}$	Koordinaten eines i -ten Sensors und eines Objekt im Fahrzeugkoordinatensystem
x_v, y_v, z_v	v -te Position
x_F, y_F, z_F $x_{Obj}, y_{Obj}, z_{Obj}$ $x_{Ref}, y_{Ref}, z_{Ref}$	Koordinaten eines mobilen Fahrzeugs, eines Objekt und einer Referenzmarke im Weltkoordinatensystem
$x_{\square a}, x_{\square l}, x_{\square r}, x_{\square e}$	Anfangs-, linker Eckpunkt-, rechter Eckpunkt- und Endwert einer trapezförmigen Zugehörigkeitsfunktion μ_\square
x_W, y_W, z_W, x, y, z	Weltkoordinaten
$\mathbf{y}(\mathbf{x})$	Übertragungsfunktionssystem
y_1, y_2	Stellgrößen eines Regelkreises bzw. des Fuzzy-Logic-Reglers
$y_{1S}, y_{2S}, y_{1Si}, y_{2Si}$	Gesamt- bzw. i -ter Teilschwerpunkt der resultierenden Zugehörigkeitsfunktionen $\mu_{1maxmin}$ und $\mu_{2maxmin}$

Verwendete Abkürzungen und Begriffe

2D	2-dimensional (x, y)
3D	3-dimensional (x, y, z)
3D-Pose	3-dimensionale Pose (x, y, α_z)
6D-Pose	6-dimensionale Pose ($x, y, z, \gamma_x, \beta_y, \alpha_z$)
Abb.	Abbildung
AGV	Automated Guided Vehicle, Autonomous Ground Vehicle
ACPI	Advanced Configuration and Power Interface
AKF	Autokorrelationsfunktion
AMR	Autonome mobile Robotik/Roboter
ASCII	American Standard Code for Information Interchange
Baud	Übertragungsrate Symbole/s
BGS	Berufsgenossenschaften
Bsp., bspw.	Beispiel, beispielsweise
BSS	Basic Service Set
ca.	circa
CCD	Charge-Coupled Device
CGI	Common Gateway Interface
CMOS	Complementary Metal Oxide Semiconductor
CW	Continuous-Wave Modulation
d.h.	das heißt
DIN	Deutsches Institut für Normung e.V.
DGPS	Differential Global Positioning System
DOS	Disk Operating System
DSL	Digital Subscriber Line
EDGE	Enhanced Data-Rates for GSM Evolution
EKF	Extended Kalman Filter
FD	ferne Distanz
fps	frames per second
FTF	Fahrerloses Transportfahrzeug
FTM	Führerlose Transportmittel
FTP	File Transfer Protocol
FTS	Fahrerloses Transportsystem

GD	Grenzdistanz
ggf.	gegebenenfalls
GIF	Graphics Interchange Format
GPIO	General Purpose Input/Output
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global Standard for Mobile Communications
GUI	Graphical User Interface
GUV	Guided Unmanned Vehicle
(h)	horizontal
HAYAI	Highspeed And Yet Accurate Indoor/outdoor-tracking
HSDPA	High Speed Downlink Packet Access
HSUPA	High Speed Uplink Packet Access
html	Hypertext Markup Language
http	Hypertext Transfer Protocol
i.A.	im Allgemeinen
I/O	Input/Output
IBSS	Independent Basic Service Set
ICP	Iterative Closest/Corresponding Points
ID	Identifikationsnummer
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IP67	IP (International Protection, Ingress Protection) -Schutzklasse 67
IR	Infrarot
ISL	Intersatellite Link
JPEG	Joint Photographic Experts Group
JPG	Joint Photographic Experts Group
Kap.	Kapitel
KF	Kalman Filter
kk, kkf	Kreuzkorrelation, Kreuzkorrelationsfunktion
KKS	Kamera-Koordinatensystem
l.	links
LAN	Local Area Network
LED	Light Emitted Diode

LF	langsam vorwärts fahren
LLL	leicht links lenken
LM	Landmarken
LRL	leicht rechts lenken
LTE	Long Term Evolution
LZW	Lempel-Ziv-Welch-Algorithmus
M.	Mitte
Matlab	Matrix Laboratory
max.	maximal, Maximum
MD	mittlere Distanz
Merlin3D	Mobile Experimental Robots for Locomotion and Intelligent Navigation with 3D PMD-camera
min.	minimal, Minimum
ML	mittelinks
MOSRO	MOBiles Sicherheits-ROboter-System
MR	mittlerechts
ND	nahe Distanz
NF	normal vorwärts fahren
NULL	nicht lenken
o.g.	oben genannt
OSI	OSI (Open Systems Interconnection Reference Model) -Referenzmodell
PCI	Peripheral Component Interconnect
PDA	Personal Digital Assistant
PG	Photo-Gate
PI	Proportional-Integral
PID	Proportional-Integral-Differential
PMD	Photonic Mixer Device
PN	Pseudo-Noise Modulation
ppr	Pulses per Rotation
PSD	Position Sensitive Device
PT1	Proportional Time 1
PT2	Proportional Time 2
PWM	Pulsweitenmodulation
r.	rechts

RF	rückwärts fahren
RFID	Radio Frequency Identification
RGB	Rot-Grün-Blau
RKS	Roboter-Koordinatensystem
RS232	Recommended Standard (früher: Radio Sector)
s.	siehe
s.a.	siehe auch
s.o.	siehe oben
s.u.	siehe unten
SF	schnell vorwärts fahren
SLAM	Simultaneous Localization and Map Building
sog.	so genannt
STOP	Stopp
Std.	Stunde
STD	Suspend to Disk
STM	Suspend to Memory
STR	Suspend to RAM
Tab.	Tabelle
TDP	Thermal Design Power
TOF	Time-of-Flight
Tom3D	Tele Operated Machine with 3D PMD-Camera
TTL	Transistor-Transistor-Logik
TÜV	Technischer Überwachungsverein
u.	und, unten
u.a.	unter anderen
u.g.	unten genannt
UD	unterschriftene Distanz
UMTS	Universal Mobile Telecommunications System
URI	Universal Resource Identifier
URL	Uniform Resource Locators
URN	Uniform Resource Names
USB	Universal Serial Bus
usw.	und so weiter
(v)	vertikal

VDI	Verein deutscher Ingenieure
VdS	Verein deutscher Sachversicherer
VGA	Video Graphics Array
VPL	Microsoft Visual Programming Language
WDS	Wireless Distribution System
WiMAX	Worldwide Interoperability for Microwave Access
Win XP	Windows eXPerience
WKS	Welt-Koordinatensystem
WL	weit links
WLAN	Wireless Local Area Network
WLL	weit links lenken
WOL	Wake on LAN
WOWL	Wake on WLAN
WR	weit rechts
WRL	weit rechts lenken
WWW	World Wide Web
XP	Windows eXPerience
z.B.	zum Beispiel

1 Einleitung

1.1 Überblick

Bereits in der 1942 erschienenen Erzählung „I, robot“ [1] veröffentlichte der Schriftsteller Isaac Asimov erstmals seine drei „Gesetze der Robotik“, in denen Verhaltensregeln von Robotern gegenüber Menschen beschrieben wurden. Später ergänzte er diese in „Robots and Empire“ [2] mit dem sog. „Nullten Robotergesetz“:

0. *Ein Roboter darf die Menschheit nicht verletzen oder durch Passivität zulassen, dass die Menschheit zu Schaden kommt.*
1. *Ein Roboter darf keinen Menschen verletzen oder durch Untätigkeit zu Schaden kommen lassen, außer er verstöße damit gegen das nullte Gesetz.*
2. *Ein Roboter muss den Befehlen der Menschen gehorchen - es sei denn, solche Befehle stehen im Widerspruch zum nullten oder ersten Gesetz.*
3. *Ein Roboter muss seine eigene Existenz schützen, solange dieses sein Handeln nicht dem nullten, ersten oder zweiten Gesetz widerspricht.*

Damals wie heute ist die Etablierung dieser Gesetze unerreicht geblieben. Nur in der Science-Fiction-Literatur und -Verfilmung verfügen Roboter scheinbar über eine Sensorik und „maschinelle Intelligenz“, die es ihnen ermöglicht, die obigen Gesetze zu beachten. Bei den heutigen militärischen Roboteranwendungen, wie Spährobotern (s. **Abb. 1.3.c**), mobilen Robotern mit Bewaffnung, Flugdrohnen, usw. bleiben die Asimovschen Gesetze bewusst unberücksichtigt oder werden nur zur Vermeidung des sog. „Friendly Fire“ mittels Freund-Feind-Erkennung beachtet.

Fast 70 Jahre nach der Veröffentlichung konnte das erste Asimovsche Robotergesetz nur teilweise innerhalb der Handhabungsrobotik und in der zivilen mobilen Robotik zur Anwendung kommen, wobei dieses Gesetz heute als Sicherheitsmaßnahme zu interpretieren und eher in Hardware als in Software implementiert ist. Der Arbeitsbereich eines Handhabungsroboters wird beispielsweise mechanisch so abgesperrt, dass Personen während der Arbeitsphase keinen Zugang dazu haben. Werden mechanische Absperrungen entfernt oder überschritten, so wird dies durch entsprechende Sensorik detektiert und der Handhabungsroboter wird deaktiviert.

Die meisten Anwendungen im Bereich der mobilen Robotik findet man als sog. Fahrerlose Transportsysteme (FTS⁴, s. **Abb. 1.1.a**), die in der Produktion für den Transport von Materialien und Produktteilen eingesetzt werden. FTS orientieren sich dabei an den in die oder auf die Fahrbahn gelegten Leitspuren (Induktionsleitungen oder Farbmarkierungen). Neben Sensoren zum Aufspüren dieser Leitspuren verfügt ein FTS auch über Sensoren zum Überwachen des Fahrwegs in seinem Frontbereich. Derzeit werden dafür mechanische Berührungssensoren (Bumper) und 2D-Laserscanner verwendet. Erfassen die Sensoren ein auf dem Fahrweg befindliches Hindernis, so wird lediglich der Fahrbetrieb des FTS unterbrochen.

Neuere Entwicklungen, wie Staubsaugerroboter, Rasenmäherroboter, Fußballroboter, Museums-, Bewachungsroboter, usw. (s. **Abb. 1.1** und **Abb. 1.2**) verfügen über mechanische Berührungssensoren (Bumper) und/oder Infrarot-Distanzsensoren zur Überwachung des nahen Fahrwegbereichs. Bei Erfassung eines Hindernisses wird der Fahrbetrieb ebenfalls unterbrochen, danach wird jedoch eine definierte Richtungsänderung vorgenommen, um schließlich den Fahrbetrieb wieder fortzusetzen.

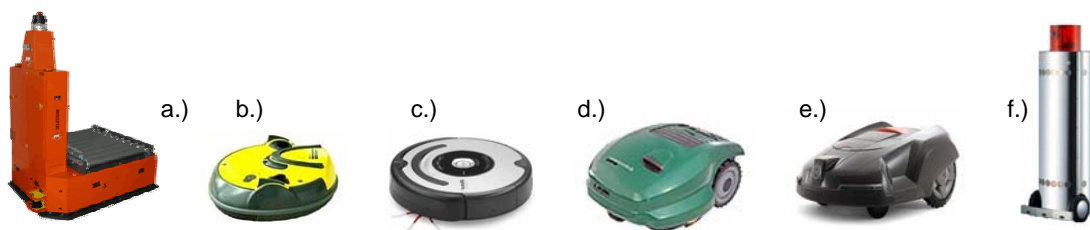


Abb. 1.1: Mobile Roboter: a.) Fahrerlose Transportsysteme (FTS) [3], b.) Robocleaner [4], c.) Roomba [5], d.) Robomow [6], e.) Automower [7] und f.) Mosro [8].

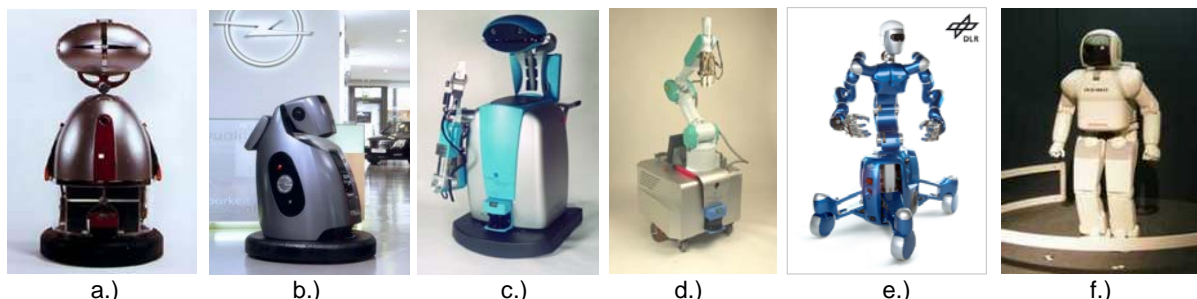


Abb. 1.2: Weitere Service-Roboter: a.) Museum Robot [9], b.) OPEL Robot [9], c.) Care-O-Bot [9], d.) Rob-at-Work [9], e.) Justin [10] und f.) Asimo [11].

⁴ Andere Bezeichnungen für FTS sind: FTM = Führerlose Transportmittel, FTF = Führerlose Transportfahrzeug, GUV = Guided Unmanned Vehicle, AGV = Automated Guided Vehicle, AGV = Autonomous Ground Vehicle, usw.

So genannte Assistenzrobotersysteme verfügen über einen Rad- oder Kettenantrieb und einen Handhabungsroboterarm (s. **Abb. 1.2.c, 1.2.d, 1.3.a** und **1.3.b**) [12] [13] [14]. Mit diesem Aufbau stellen die Systeme den Übergang von mobilen Robotern hin zu Humanoiden dar. Ein typischer Humanoid ähnelt im Aufbau einem Menschen und besteht aus einem Schreitantrieb mit zwei Beinen, einem Rumpf, zwei Handhabungsroboterarmen und einem Stereovision-System als Kopf (s. **Abb. 1.2.f**). Für Aufgaben mit begrenztem Aktionsbereich kann ein Humanoid auch ohne Schreitantrieb ausgestattet sein und ähnelt damit einem Industrieroboter, der direkt auf festem Grund fixiert ist. Statt auf festem Grund kann eine solche Konstruktion auch auf einer beweglichen Plattform fixiert und mit Rad- oder Kettenantrieb versehen sein (s. **Abb. 1.2.e**).

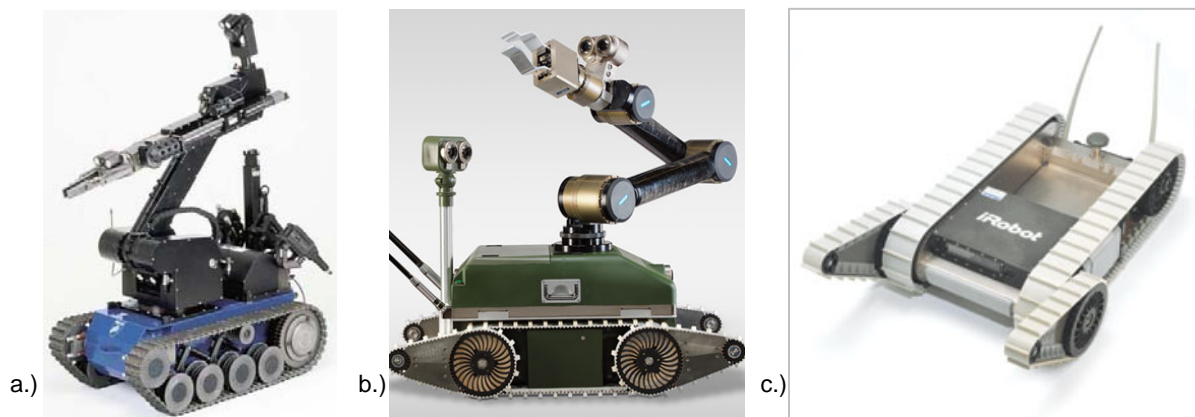


Abb. 1.3: Mobile Roboter mit Kettenantrieb: a.) „tEODor“ [15], b.) „Asendro-Scout“ [8] und c.) „PackBot“ [5].

Eine besondere Beachtung gilt dem mobilen Fahrzeug „Stanley“ von Prof. Thrun [237] [249] (Stanford University, USA), einem umgebauten Landfahrzeug VW-Touareg (s. **Abb. 1.4.a**), welches im Rahmen des DARPA Grand Challenge [16] in 2005 zwischen den Städten Los Angeles (USA) und Las Vegas (USA) innerhalb von 10 Stunden ca. 320 km Entfernung autonom navigieren konnte. Dieses Fahrzeug wurde mit Radarsensor, Laserscanner und Stereo-Kamerasystem ausgestattet, zwecks Geländeüberwachung, Objekt- und Hinderniserkennung. Für die Positions- und Orientierungsbestimmung kamen ein Trägheitssensor, mehrere Rad-Encoder und ein GPS-Sensor zum Einsatz. Das mobile Fahrzeug „Stanley“ konnte hierbei den ersten Platz belegen. Im Rahmen des DARPA Urban Challenge [17] in 2007 erreichte dieses mobile Fahrzeug den zweiten Platz. Hier sollte das mobile Fahrzeug innerhalb eines bebauten Gebietes auf dem ehemaligen Luftwaffenstützpunkt „George Air Force Base, USA“ autonom navigieren.

Zu den populärsten Vertretern der mobilen Robotik zählen die Mars-Exploration-Roboter Pathfinder [18] [112], Spirit [19], Opportunity [19] und der am 26.11.2011 gestartete Curiosity [20] (**Abb. 1.4.b**) von der NASA sowie der Mars-Exploration-Roboter ExoMars (Missionsstart voraussichtlich in 2016, s. **Abb. 1.4.c**) von der ESA [21] [22] [113]. Zur Hinderniserkennung verfügen diese Roboter je über ein Stereovision-System, welches u.a. den Fahrweg in 3-dimensionaler Form überwachen.



a.)



b.)



c.)

Abb. 1.4: a.) Mobiles Fahrzeug Stanley [16], b.) Curiosity ($v_F = 6 \text{ m/Std.}$) [20] der NASA und c.) Exomars [21] [22] [113] der ESA.

Abgesehen vom ersten Asimovschen Robotergesetz werden alle übrigen von den o.g. Robotersystemen nicht erfüllt. Dabei mangelt es nicht an geeigneter Sensorik und Aktorik, sondern es fehlt an schneller Datenerfassung und -interpretation der Umgebung in Echtzeit. Man würde dafür ein System benötigen, welches beispielsweise eine Person als solche erfassen, deren Handlungen in Interaktion mit ruhenden sowie beweglichen, jeweils festen wie auch formveränderlichen Objekten erkennen, daraus Gefahrensituationen identifizieren oder gar vorausberechnen und schließlich entsprechend darauf reagieren könnte. Was bei einfachsten biologischen Systemen funktioniert, z.B. die Reaktion auf Umgebungsreize bei Insekten, ist bei Robotersystemen nur in simpler Form implementiert.

Dennoch hat sich die Robotik seit Asimovs Veröffentlichungen zu einer technisch und wissenschaftlich anspruchsvollen Disziplin entwickelt. Aus ökonomischen Gründen wurde die Weiterentwicklung insbesondere der Handhabungsrobotik und FTS zwecks Steigerung der industriellen Produktivität vorangetrieben. Starke Impulse kommen heute auch aus Dienstleistung, Sicherheitstechnik, Raumfahrt und Militär. Im Gegensatz zur Handhabungsrobotik und zum FTS müssen solche Systeme nicht nur im Innenbereich eines Gebäudes, sondern auch im freien unebenen Gelände

agieren können. Spätestens hier wird die Problematik einer 3D-Umgebungsmodellierung und einer 6D-Selbstlokalisierung deutlich.

Neben dem eigentlichen Einsatzgebiet beeinflussen sowohl die Umgebung als auch die gewünschte, durchzuführende Dienstleistung die Architektur solcher Robotersysteme. Beispielsweise reicht zur Durchführung von Transportaufgaben die Verwendung des Fahrwerks selbst aus. Je nach Eigenschaften der Umgebungen können dafür Rad-, Ketten- oder auch Schreitantriebe eingesetzt werden. Für komplexe Dienstleistungen muss neben dem Fahrwerk zusätzliche Aktorik verbaut sein. Möglich wären z.B. ein Mähwerk, Staubsauger, einfacher Greifer, ein Handhabungsroboterarm mit Greifer, usw.

1.2 Problemstellung der Arbeit

Die Bedeutung der räumlichen Datengewinnung wird in biologischen Systemen von den Insekten mit ihren Facettenaugen bis hin zu den Säugetieren mit ihren Aktiv-Stereo-Augenpaaren eindringlich demonstriert. Diese räumliche Datengewinnung umfasst hier hauptsächlich die 6D-Eigenbewegung des Individuums selbst, die Erkennung von Objekten sowie deren Bewegungen und die 3D-Umgebungsmodellierung.

In technischen Anwendungen wird derzeit die räumliche Datengewinnung hauptsächlich in der visuellen Qualitätssicherung, der Handhabungsrobotik und in der mobilen Robotik benötigt. Die Erfassung der 3D-Umgebung bzw. der 6D-Pose in Echtzeit sind mittels bildgebender Sensorik möglich. Als bildgebende Sensoren kommen hierfür die Mono-Kamera bzw. 2D-Kamera, omnidirektionale 2D-Kamera, Motion-Stereo-Kamera mit nur einer 2D-Kamera, passive sowie aktive Stereovision mit zwei 2D-Kameras, 2D-Laserscanner sowie 3D-Laserscanner zum Einsatz [23] [24].

Eine 2D-Kamera liefert im Allgemeinen ausschließlich 2D-Videobilder ohne Tiefeninformationen. Durch z.B. die parallele Anordnung zweier 2D-Kameras als Stereo-Kamerasystem können diese Tiefeninformationen nur teilweise mit hohem rechnerischen Aufwand rekonstruiert werden. Dagegen kann der Laserscanner

grundsätzlich nur Tiefeninformationen der Umgebung generieren. Da jedoch beim Scan-Vorgang stets mechanische Komponenten innerhalb des Laserscanners beteiligt sind, können nur geringe Frame-Raten erzielt werden.

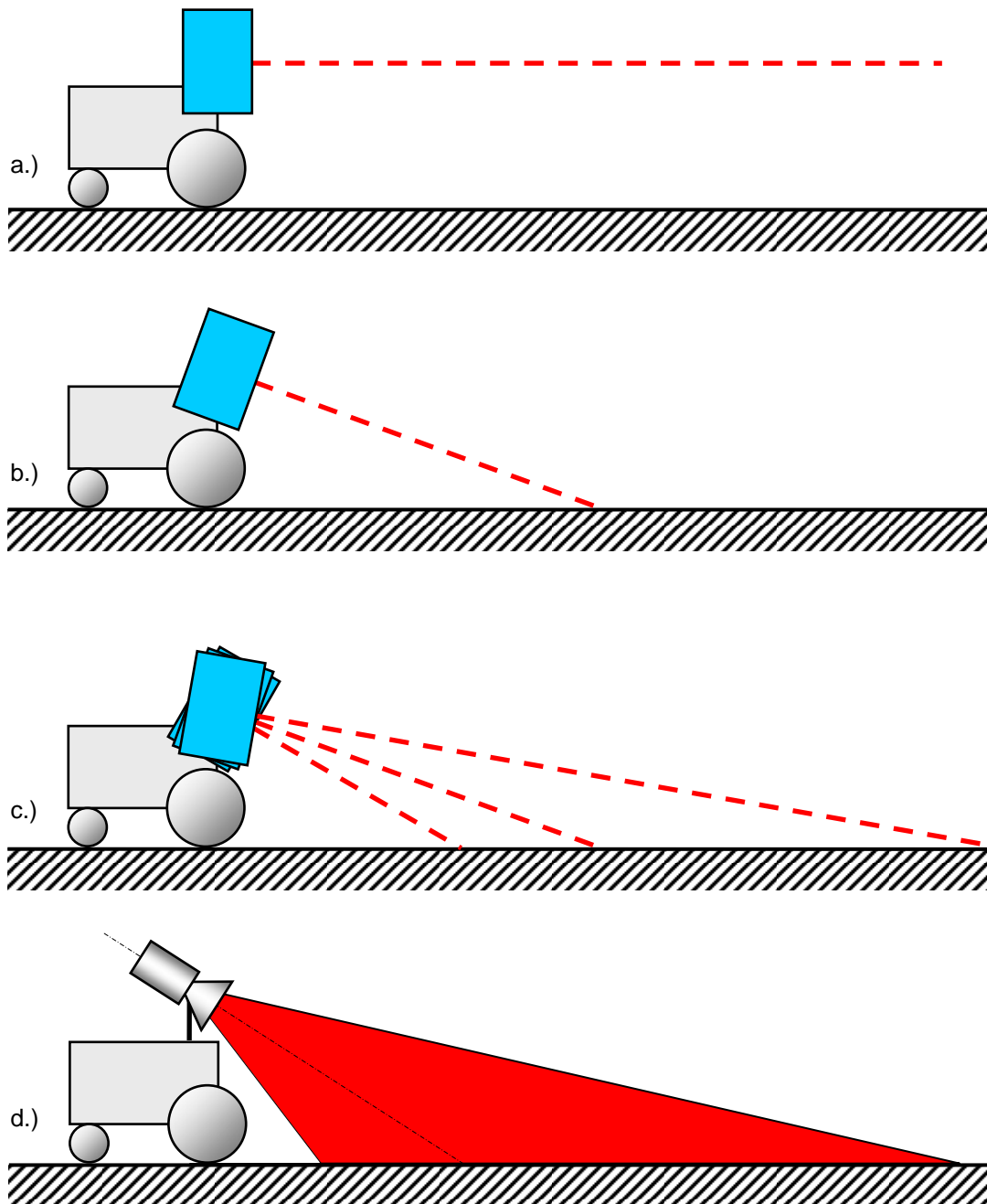


Abb. 1.5: Prinzip der Erfassung von Hindernissen im Frontbereich eines mobilen Fahrzeugs mit einem a.) 2D-Laserscanner, b.) geneigten 2D-Laserscanner, c.) 3D-Laserscanner und d.) mit einer PMD-Kamera.

Dennoch wird der Laserscanner wegen seiner Fähigkeit, weite Distanzen messen zu können - und dies bei relativ geringem Datenumfang pro Scan - sowie aufgrund des weiten optischen Öffnungswinkels in vielen Roboteranwendungen erfolgreich einge-

setzt [25] [26]. In **Abb. 1.5** kann er als 2D-Laserscanner mit nur einer einzelnen Erfassungsebene wirken. Bei einem geneigten 2D-Laserscanner sorgt die zusätzliche Fahrbewegung des mobilen Fahrzeugs für eine 3-dimensionale Datenerfassung. Mit einem 3D-Laserscanner kann die Umgebung 3-dimensional abgetastet werden [27].

1.3 Motivation

Ein 3D-Laserscanner nimmt ein Scan-Bild mit 16.200 Punkten in ca. 1,35 Sekunden auf [28]. Eine PMD-Kamera hingegen, z.B. die PMD [vision] CamCube in **Abb. 1.6** r., kann ein PMD-Videobild⁵ mit 40.000 PMD-Pixel in 0,025 Sekunden aufnehmen. Damit besitzt die PMD-Kamera mit 40 fps gegenüber dem 3D-Laserscanner mit 0,74 fps eine deutlich höhere Frame-Rate, was Voraussetzung für die Navigation eines mobilen Fahrzeugs in Echtzeit ist.

Mit einem optischen Öffnungswinkel von 180° (h) und 120° (v) eignet sich der 3D-Laserscanner eher für die 3D-Umgebungsmodellierung, wobei die geringe Frame-Rate einen Echtzeitbetrieb erschwert. Demgegenüber besitzt die PMD-Kamera einen eher kleinen optischen Öffnungswinkel von 40° (h) und 40° (v). Für die Hindernis- und Fahrwegerkennung eines mobilen Fahrzeugs mit Hilfe der PMD-Kamera erweist sich dies jedoch als vorteilhaft, da der Frontbereich des mobilen Fahrzeugs weitestgehend durch diesen optischen Öffnungswinkel abgedeckt und dieser Bereich mit einer hohen Auflösung von 200x200 PMD-Pixel erfasst wird. Zukünftig kann mit leistungsfähigeren PMD-Kameras gerechnet werden, z.B. mit > 50 fps, 640x480 PMD-Pixel (VGA-Format) und mit einem optischen Öffnungswinkel > 40°.



Abb. 1.6: PMD (Photonic Mixer Device)-Kameras: PMD [vision] 3k-S mit 64x48 PMD-Pixel (l.) und die PMD [vision] CamCube 3 mit 200x200 PMD-Pixel (r.).

⁵ Bedeutungsgleich mit PMD-Videodaten, PMD-Tiefenwertbild, PMD-Entfernungswertbild.

Die maximal erfassbare Distanz der PMD-Kamera mit $f_m = 20$ Mhz (s. Kap. 2.3) liegt bei 7,5 m, wodurch die Ausdehnung des zu überwachenden Frontbereichs des mobilen Fahrzeugs entsprechend begrenzt wird (s. a. **Abb. 1.5.d**). Dennoch soll bei einer Fahrzeuggeschwindigkeit von ca. 1 m/s eine vorausschauende Überwachung des Frontbereichs des mobilen Fahrzeugs gewährleistet sein.

Mit diesen derzeit verfügbaren Eigenschaften der PMD-Kamera soll hier nun deren Einsatz in der mobilen Robotik insbesondere für die Hindernis- und Fahrwegerkennung in Verbindung mit der Hindernisvermeidung sowie für die Selbstlokalisierung näher untersucht werden.

1.4 Aufbau der Arbeit

Der Einsatz der PMD-Kamera in der mobilen Robotik einschließlich FTS soll hier exemplarisch gezeigt werden. Dies umfasst zunächst die Beschreibung der PMD-Kamera selbst (s. Kap. 2). Deren Integration in ein mobiles Fahrzeug sowie die Hindernis- und Fahrwegerkennung mit Hilfe der PMD-Kamera erfolgt in Kap. 3. Nach der Beschreibung der beiden mobilen Roboter Tom3D und Merlin3D in Kap. 4 folgt die lokale Navigation in Kap. 5 und die Selbstlokalisierung in Kap. 6 unter Verwendung der PMD-Kamera.

Die globale Navigation ist nur dann realisierbar, wenn dem mobilen Fahrzeug entsprechende teilautonome Funktionen als Handlungsanweisungen zur Verfügung stehen. Aus der Kombination solcher teilautonomen Funktionen können neue höherwertige Funktionen kreiert werden, die hier als Dienstleistung eines mobilen Service-Roboters aufgefasst werden können. Diese zukünftig für die Robotik bedeutsame Komponente der Erstellung von Dienstleistungen wird beispielhaft in Kap. 7 aufgezeigt.

1.5 Wissenschaftlicher Beitrag

Auf der Basis von PMD-Videodaten der PMD-Kamera wird in dieser Arbeit eine schnelle 3D-Fahrweg- und 3D-Hinderniserkennung sowie eine Fuzzy-Logic-Regelung zur Hindernisvermeidung entwickelt, die sowohl positive als auch negative Hindernisse berücksichtigt. Auch ohne Selbstlokalisierung sind damit bereits

einfachste Dienstleistungen möglich, wie z.B. Suchaufgaben mittels eines mobilen Fahrzeugs.

Poseänderungen eines mobilen Fahrzeugs, d.h. Änderungen von Position und Orientierung, können in einer PMD-Videobildfolge als Verschiebungen von Objekten und deren markanten Merkmalen festgestellt werden. Diese Merkmale gilt es innerhalb eines PMD-Videobildes zu finden und selbige im darauf folgenden PMD-Videobild wieder zu entdecken. Aus den Verschiebungen der Merkmale kann schließlich der Transformationsvektor für 3D-Fahrbewegungen mit Hilfe der Einheitsquaternionen-Methode und für 2D-Fahrbewegungen mit Hilfe der HAYAI (Highspeed And Yet Accurate Indoor/outdoor-tracking)-Matching-Methode ermittelt werden. Der so ermittelte Transformationsvektor ist mit Fehlern behaftet, die mit Hilfe der ICP (Iterative Closest Points)-Methode reduziert werden können. Alle drei Methoden wurden bereits bei Laserscanner-Anwendungen eingesetzt [29] [199] [200], sollen hier nun auf PMD-Videobilder, unter Nutzung konventioneller Bildverarbeitungsalgorithmen [45] [135], angewandt werden.

Der Transformationsvektor entspricht genau dem Bewegungsvektor des mobilen Fahrzeugs, welcher die Bewegungstrecke und -richtung innerhalb der Zeit zwischen zwei aufeinander folgenden PMD-Videobildern angibt. In Verbindung mit der Koppelnavigation, also der Aufaddierung aller Transformations- bzw. Bewegungsvektoren seit Fahrbeginn, erfolgt daraus die Selbstlokalisierung des mobilen Fahrzeugs.

2 Die PMD-Kamera

Abgesehen von Ultraschall- und IR-Sensoren werden für die 3D-Umgebungserfassung innerhalb der mobilen Robotik vor allem 2D- und 3D-Laserscanner oder auch Stereo-Kamerasysteme mit 2D-Kameras als bildgebende Sensoren eingesetzt:

- Ein Laserscanner [25] [26] [27] erfasst die Umgebung eines mobilen Roboters lediglich in einer Ebene, d.h. es liegen damit nur 2D-Tiefendaten vor. Dabei wird ein einzelner Laserstrahl über einen mechanisch bewegten Spiegel periodisch abgelenkt. Durch zusätzliche mechanische Schwenk- bzw. Nickbewegungen des gesamten Laserscanner-Systems können mehrere Ebenen der Roboterumgebung erfasst werden, so dass diese nun in 3D-Tiefendaten vorliegen [28] [29] [30] [31]. Aufgrund der mechanischen Ablenkbewegungen innerhalb eines sog. 3D-Laserscanners benötigt die Erfassung eines einzelnen 3D-Bildes bei einem Scan-Winkelbereich von $180^\circ \times 120^\circ$ und bei reduzierter Auflösung ca. 1,35 Sekunden [28].
- Ein Stereo-Kamerasystem besteht aus zwei in definiertem Abstand nebeneinander und parallel zu ihren optischen Achsen fixierten 2D-Kameras (CMOS- oder CCD-Technologie), aus deren Stereo-Doppelbildern die Tiefeninformation für jedes Pixel errechnet werden kann. Dabei wird innerhalb des Stereo-Doppelbildes nach korrespondierenden 2D-Bildelementen gesucht, z.B. hinreichend grobe Textur, um z.B. über das Verfahren der flächenbasierten Korrelation (s. a. Kap. 6.6) den Abstand zu berechnen. Zwei leistungsfähige Systeme seien hier genannt: Das Stereo-Kamerasystem „DeepSea“ der Firma Tyzx [32] erreicht bei einer Bildgröße von 512×480 Pixel und einer Genauigkeit von 52 Tiefenstufen eine 3D-Bildrate von 30 Hz. Die Fa. Point Grey Research [33] bietet das Stereo-Kamerasystem „Triclops“ an, welches bei einer Bildgröße von 640×480 Pixel eine 3D-Bildrate von 4,4 Hz besitzt. Bei reduzierter Bildgröße von nur 160×120 Pixel ist sogar ein System mit 91 Hz 3D-Bildrate erhältlich.

2.1 Varianten der PMD-Kamera und andere 3D-Kameras

Stereo-Kamerasysteme haben stets den Nachteil, dass die Aufnahmen der Fahrzeugumgebung stark von künstlichen und natürlichen externen Lichtquellen beeinflusst werden. Lichtintensität und Positionen dieser Lichtquellen können variieren und aufgrund des veränderten Schattenwurfs entsprechende Fehler bei der Zuordnung korrespondierender 2D-Bildelemente innerhalb der Stereo-Doppelbilder hervorrufen. Demgegenüber besitzt die PMD-Kamera ihre eigene Beleuchtungseinheit. Mit Hilfe der Fremdlichtunterdrückung kann der Einfluss externer Lichtquellen nahezu eliminiert werden. Folgende mit Fremdlichtunterdrückung ausgestattete PMD-Kameras seien genannt [34]:

- Die PMD [vision] 3k-S (**Abb. 2.1.a**) verfügt über 64x48 PMD-Pixel, einen optischen Öffnungswinkel 40° (h) und 30° (v), max. Distanz 7,5 m, Distanzauflösung > 6 mm, Frame-Rate 25 Hz und eine Fremdlichtunterdrückung. Diese 3D-Kamera wurde im Verlauf dieser Arbeit näher untersucht und beschrieben bzw. hier für den Einsatz in der mobilen Robotik verwendet.
- Die PMD [vision] O3 (**Abb. 2.1.b**) enthält 64x48 PMD-Pixel, einen optischen Öffnungswinkel 30° (h) und 40° (v), max. Distanz 7,5 m, Distanzauflösung >6mm, Frame-Rate 25 Hz und eine Fremdlichtunterdrückung. Mit der Schutzart/-klasse IP65 kann diese Kamera auch im Außenbereich zum Einsatz kommen.
- Die PMD [vision] CamCube (**Abb. 2.1.c**) besitzt 200x200 PMD-Pixel, einen optischen Öffnungswinkel 40° (h) und 40° (v), max. Distanz 7 m, Distanzauflösung < 3 mm, Frame-Rate 25 Hz und eine Fremdlichtunterdrückung. Auch diese Kamera entspricht der Schutzart/-klasse IP65 und kann damit ebenfalls für den Außenbereich eingesetzt werden.



Abb. 2.1: 3D-Kameras der Fa. PMDTec [34] mit Fremdlichtunterdrückung: a.) PMD [vision] 3k-S, b.) PMD [vision] O3 und c.) PMD [vision] CamCube.

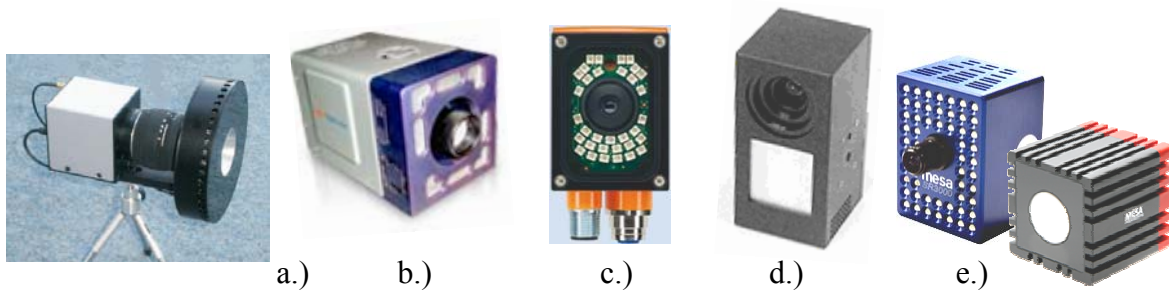


Abb. 2.2: Andere 3D-Kameras: a.) „2D/3D-Multichip-Kamera“ [35] [36], b.) „Z-Cam“ [40], c.) „Efactor PMD 3D“ [39], d.) „Canesta Vision Camera“ [41] sowie e.) „SwissRanger SR3000“ und „SwissRanger SR4000“-Kamera [42].

Die 2D/3D-Multichip-Kamera (**Abb. 2.2.a** und **2.12.b**) vom ZESS der Universität Siegen [35] [36] stellt insoweit eine Besonderheit dar, als dass sie neben einem originalen PMD-Chip zusätzlich einen CCD-Chip enthält. Beide bildgebenden Sensoren sind optisch an einen spektralen Strahlteiler [37] [38] gekoppelt. Durch die Blende und das Objektiv hindurch fällt das modulierte IR-Licht zusammen mit dem Umgebungslicht der Szene, teilt sich am spektralen Strahlteiler auf und gelangt je nach Wellenlänge in den entsprechenden Chip. Mit dieser Kamera können nun sowohl Tiefenwertbilder als auch hochauflösende 2D-Farbbilder aufgenommen werden. Trotz unterschiedlicher Auflösung der 2D- und 3D-Videobilder werden diese innerhalb dieser Kamera pixelidentisch zugeordnet.

Des Weiteren findet sich die PMD-Technologie in verschiedenen 3D-Kameras der Fa. IFM-Electronic [39] je mit einem originalen PMD-Chip wieder. Die „Efactor PMD 3D“-Kamera (**Abb. 2.2.c**) verfügt über einen 64x48 PMD-Chip, einen optischen Öffnungswinkel 30° (h) und 40° (v), max. Distanz 6,5 m, Frame-Rate 25 Hz, jedoch ohne Fremdlichtunterdrückung. Diese 3D-Kamera ist zusammen mit der Beleuchtungseinheit in einem IP67-Gehäuse untergebracht und kann damit im industriellen Bereich eingesetzt werden. Da jedoch eine Fremdlichtunterdrückung fehlt, ist dieser Typ von 3D-Kamera nur in abgeschlossenen Bereichen ohne fremde Lichtquellen geeignet.

Neben den o.g. PMD-Kameras der Fa. PMDTec [34] sind weitere 3D-Kameras auf dem Markt erhältlich, die alle ähnlich der PMD-Methode mit dem Time-of-Flight-Prinzip funktionieren (s. Kap. 2.3). Und zwar die „Z-Cam“ (**Abb. 2.2.b**) der Fa. 3DV Systems [40], die „Canesta Vision Camera“ (**Abb. 2.2.d**) der Fa. Canesta [41], sowie die „SwissRanger SR3000“ und „SwissRanger SR4000“-Kamera (**Abb. 2.2.e**) der

Fa. MESA-Imaging [42] (früher Fa. CSEM [43] [44]). Letztere besitzt 176×144 Pixel und verfügt ebenfalls über eine Fremdlichtunterdrückung.

2.2 Kalibrierung der PMD-Kamera

Wie bei einer konventionellen 2D-Kamera wird auch bei der PMD-Kamera ein optisches Linsensystem verwendet. Daher sind bei der Kalibrierung, auch Lateral-Kalibrierung genannt, die typischen intrinsischen Parameter zu ermitteln, wie Linsenverzerrung, optischer Fokus f , das tatsächliche Bildzentrum des PMD-Videobildes, usw. Diese konventionellen Kalibrierungsparameter können beispielsweise mit Hilfe der bereits verfügbaren OpenCV-Funktionsbibliothek der Fa. Intel [45] ermittelt werden [46] [47] [48] [49]. Mit Hilfe der von J. Y. Bouguet erstellten Toolbox [50] können solche Kalibrierungsarbeiten auch unter Matlab [51] durchgeführt werden.

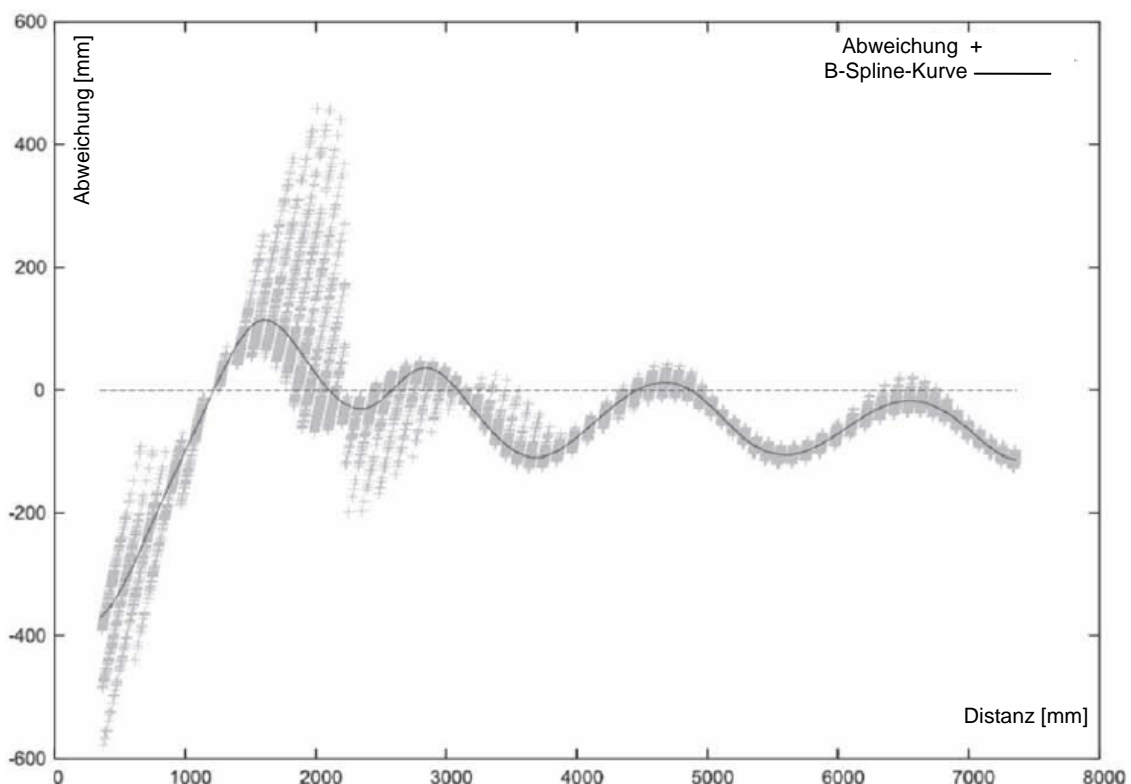


Abb. 2.3: Fehlerverlauf bei der Distanzmessung eines PMD-Pixels mit einem periodischen Fehlerverlauf, ähnlich einer B-Spline-Funktion [46] [47] [48] [49].

Statt der gewünschten distanzabhängigen linearen Charakteristik weisen die Tiefenwerte je PMD-Pixel einen periodischen Fehlerverlauf entsprechend **Abb. 2.3** auf, der mittels der Distanzkalibrierung zu kompensieren ist. Diese Distanzverzerrung kann entstehen durch:

- die Demodulation des optischen Empfangssignals innerhalb der Korrelationsfunktion eines jeden PMD-Pixels,
- geometrische Fertigungsfehler im PMD-Pixel und deren Anordnung zu einem gesamten PMD-Chip,
- Modulationsfehler im Sendesignal,
- Leuchtcharakteristik der Beleuchtungseinheit,
- Reflexionseigenschaften und Orientierung der Objektoberflächen, usw.

Zur Kompensation dieser Fehlerarten wird in Kolb et al. [46] - [49] eine Global-Distanzkalibrierung, also insgesamt alle PMD-Pixel betreffend, in Kombination mit einer zusätzlichen Pro-Pixel-Distanzkalibrierung vorgestellt. Weitere Kalibrierungs-, Kompensations- und Verbesserungsmaßnahmen betreffen die Bewegungsverwischungen (motion blurring), die Distanzdatenverfeinerung, die Konvertierung der PMD-Videobildaten von Kugel- in kartesische Koordinaten, die geometrische Rekonstruktion der im PMD-Videobild enthaltenen Objekte, usw.

Als weiterer Kameraparameter wird in Wiedemann et al. [52] der horizontale und vertikale optische Öffnungswinkel der PMD-Kamera berechnet:

$$\varphi_h = 2 \cdot \arctan\left(\frac{d_h}{2f}\right) \quad \text{und} \quad \varphi_v = 2 \cdot \arctan\left(\frac{d_v}{2f}\right) \quad (2.1)$$

Wobei d_h und d_v für die horizontalen und vertikalen Abmessungen des PMD-Chip und f für den Fokus stehen. Insbesondere für die Konvertierung des PMD-Videobildes von kugel- in kartesische Koordinaten sind diese Daten von Bedeutung.

Neben den o.g. Kalibrierungen existiert schließlich noch die Nullpunktkalibrierung. Hier kann der Koordinatenursprung des Tiefenwertbildes gezielt für die aktuelle Anwendung definiert werden. Bei Anwendungen mit feststehender PMD-Kamera kann z.B. eine im Hintergrund befindliche Wand einer aufzunehmenden Szene als Koordinatenursprung definiert werden. Bei einem mobilen System, in dem die PMD-Kamera mit einem speziellen Neigungswinkel auf den Fahrweg ausgerichtet ist (s. Kap. 3), sollte der Nullpunkt entweder im Ursprung des PMD-Kamera-Koordinatensystems oder im vordersten Frontbereich des mobilen Fahrzeugs liegen. Wird bspw. der Nullpunkt auf den mechanischen Bumper gelegt, dann kann die

tatsächliche Roboter-Hindernis-Distanz angegeben werden. Die Nullpunkt-kalibrierung kann mittels der zur PMD-Kamera gehörenden Software „CamVis Pro“ [53] [54] durchgeführt werden. Allerdings bleibt dabei der in **Abb. 2.3** dargestellte Fehlerverlauf unberücksichtigt.

2.3 Funktionsweise der PMD-Kamera

Die PMD-Kamera soll hier nun bezüglich ihrer Eigenschaften und Parameter näher betrachtet werden, die bei der Verwendung dieser Kamera zu beachten sind und deren Beschreibung über die übliche Literatur [55] - [75] hinausgeht. Dies betrifft insbesondere die für die Bildverarbeitung relevanten Details bzw. die per Software einzustellenden und abrufbaren Daten/Parameter. Beispielsweise soll der Zusammenhang von Modulationsfrequenz f_m und Eindeutigkeitsbereich l_{PMD} , Integrationszeit T_{int} und Rauschen im PMD-Videobild gezeigt werden.

Die PMD-Kamera stellt einen bildgebenden Sensor dar, der über das normale 2D-Grauwertbild hinaus auch Tiefen- bzw. Abstandsinformationen einer Szene als 3D-Videobild direkt aufnehmen kann. Hauptbestandteile der PMD-Kamera sind die externe Sendeeinheit (LED-Array) und insbesondere der interne PMD-Chip mit einer Auflösung von derzeit, je nach Ausführung, 64×16, 64×48, 160×120 oder 200×200 PMD-Pixel.

Jedes einzelne PMD-Pixel funktioniert ähnlich einem Laserradar-Empfänger nach dem Time-of-Flight (TOF, Echolaufzeit)-Prinzip, wobei die Laufzeit des modulierten Lichtsignals vom Sender (LED-Array) zum reflektierenden Objekt und zurück zum Empfänger anhand der Lichtlaufzeit pixelweise ermittelt wird. Dies bedeutet konkret, dass bei einer Distanz d das Licht mit der Lichtgeschwindigkeit c den Hin- und Rückweg von $2 \cdot d$ durchläuft und dafür die Zeit $t_{TOF} = \tau$ benötigt [55] - [75]:

$$2 \cdot d = t_{TOF} \cdot c \rightarrow d = \tau \cdot c / 2 \quad (2.2)$$

Aufgrund der sehr kleinen Lichtlaufzeiten τ können diese nur mit aufwendigen und komplexen Maßnahmen ermittelt werden. Abhilfe schafft hier die Messung der Phasendifferenz $\Delta\varphi$ zwischen gesendetem und empfangenem Lichtsignal. Folgende

Verhältnisgleichung gibt die Beziehung zwischen Phasendifferenz $\Delta\varphi$ und Lichtlaufzeiten τ wieder:

$$\frac{\Delta\varphi}{2\pi} = \frac{\tau}{T_m} \quad (2.3)$$

Mit $T_m = 1/f_m$ und $f_m =$ Modulationsfrequenz ergibt sich damit eine Lichtlaufzeit von

$$\tau = \frac{\Delta\varphi}{2\pi \cdot f_m} \quad (2.4)$$

Das gesendete Licht wird also am Objekt reflektiert und durch die unterschiedlichen Raumtiefen d der Objektkontur entstehen unterschiedliche Lichtlaufzeiten τ . Das reflektierte Licht wird dann via Kameraoptik auf den PMD-Chip als Interferenzmuster projiziert, welches in jedem PMD-Pixel ausgewertet und eingelesen wird. Pro PMD-Pixel wird also die Phasendifferenz $\Delta\varphi$ zwischen gesendetem Modulationssignal $u_m(t)$ und empfangenem Lichtsignal $P_{opt}(t)$ während der Integrationszeit T_{int} mit folgender Beziehung gemessen:

$$kkf(\tau) = \frac{1}{T_{int}} \cdot \int_{t=0}^{T_{int}} P_{opt}(t) \cdot u_m(t + \tau) dt \quad (2.5a)$$

$$\text{mit } T_{int} = n \cdot T_m \text{ und } n = 1, 2, 3, \dots [65] \quad (2.5b)$$

Gleichung (2.5a) stellt eine Kreuzkorrelationsfunktion dar und setzt sich aus einer Multiplikation und einer Integration zusammen. Da in (2.5a) das gesendete mit dem empfangenen Lichtsignal korreliert wird, d.h. beide Signale besitzen die gleiche Signalform, spricht man hierbei von einer Autokorrelationsfunktion (AKF). Aus nachrichtentechnischer Sicht arbeitet das PMD-Pixel entsprechend einem Korrelationsempfänger, der Mischung, Abtastung und Integration beinhaltet.

Bei der Modulationsart des gesendeten Infrarotlichts unterscheidet man zwischen Pulsmodulation und CW (Continuous Wave)-Modulation, wobei letztere die Sinus-, Rechteck- und PN (Pseudo-Noise)-Modulation beinhaltet [74]. Die Kreuzkorrelationsfunktion $kkf(\tau)$ in (2.5a) sei am Beispiel eines Rechtecksignals mit Periodendauer T_m bei CW-Modulation und mit 50%-igem Tastverhältnis in **Abb. 2.4** demonstriert:

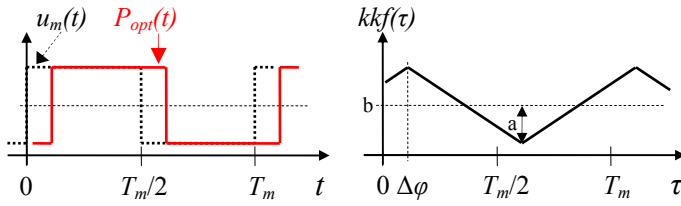


Abb. 2.4: $kkf(\tau)$ eines Rechtecksende- $u_m(t)$ und Rechteckempfangssignals $P_{opt}(t)$ bei CW-Modulation.

Die $kkf(\tau)$ in **Abb. 2.4** stellt eine Dreiecksfunktion dar, die jedoch in der Praxis eher einer Dreiecksfunktion mit abgerundeten Spitzen ähnelt, da die Flanken insbesondere des Empfangssignals $P_{opt}(t)$ durch Umgebungseinflüsse, wie unterschiedliche Reflexionen und Absorptionen, abgeflacht werden.

Anhand der $kkf(\tau)$ kann nun die Phasendifferenz $\Delta\varphi$ ermittelt werden. Hierfür reicht es aus, vier Abtastwerte $A_1 = kkf(\tau=0)$, $A_2 = kkf(\tau = T_m /4)$, $A_3 = kkf(\tau = T_m /2)$ und $A_4 = kkf(\tau = 3 T_m /4)$ der Kreuzkorrelationsfunktion entsprechend **Abb. 2.4** zu entnehmen, die jeweils 90° zueinander verschoben sind [60] [65]. Diese Abtastwerte A_1, A_2, A_3 und A_4 dienen dann zur Berechnung der Phasendifferenz $\Delta\varphi$, der Amplitude a und des Offset b :

$$\Delta\varphi = \arctan\left(\frac{A_1 - A_3}{A_2 - A_4}\right) \quad (2.6)$$

$$a = \frac{1}{2}\sqrt{(A_1 - A_3)^2 + (A_2 - A_4)^2} \quad (2.7)$$

$$b = \frac{1}{4}(A_1 + A_2 + A_3 + A_4) \quad (2.8)$$

Der Offset b stellt hierbei die Helligkeit bzw. den Grauwert pro PMD-Pixel dar und bildet zusammen mit allen anderen PMD-Pixeln das Grauwertbild der PMD-Kamera. Die Amplitude a der Kreuzkorrelationsfunktion pro PMD-Pixel wird hauptsächlich von der Signalamplitude des reflektierten Lichtsignals $P_{opt}(t)$ beeinflusst und gibt Auskunft über die Genauigkeit des ermittelten Distanzwertes d [75]. D.h. zur Beurteilung der Genauigkeit des Distanzwertes eines PMD-Pixels, kann stets die entsprechende Amplitude a herangezogen werden. Beispielsweise bei der Hinderniserkennung eines mobilen Fahrzeugs kann bei zu kleiner Amplitude a bzw. bei zu geringer Genauigkeit der Distanzwert des entsprechenden PMD-Pixels verworfen werden. Dies ist insbesondere dann der Fall, wenn Hindernisse mit einer Distanz detektiert werden, die über den Eindeutigkeitsbereich hinaus reichen. Aus der Phasendifferenz $\Delta\varphi$, sowie den Gleichungen (2.2) und (2.4) lässt sich schließlich die Distanz d pro PMD-Pixel wie folgt berechnen:

$$d = \frac{c \cdot \Delta\varphi}{4\pi \cdot f_m} \quad (2.9)$$

2.3.1 Integrationszeit

Ein PMD-Pixel stellt im Prinzip zwei in Sperrrichtung betriebene Dioden dar, zwischen denen Kathoden zwei zusätzliche Kontakte für die Modulationseingänge u_{ma} und u_{mb} des gesendeten Lichtsignals u_m , integriert sind (s. **Abb. 2.5**) [65]. Beide Kontakte sind lichtdurchlässig, daher auch Photo-Gates genannt, durch die das empfangene Lichtsignal P_{opt} fällt.

Die Kathoden K_a und K_b , sowie weitere hier nicht gezeigte Kontakte, sind jeweils mit definierten Potentialen „vorgespant“ und bilden somit die in **Abb. 2.6.a** und **2.6.b** dargestellten Potentialverläufe innerhalb des PMD-Halbleitersubstrats. Die Potentialverläufe in **Abb. 2.6.c** bis **2.6.f** stellen sich bei Gegentaktmodulation ein, d.h. bei CW-Modulation z.B. mit Rechtecksignal liegen im Zeitbereich $0 \leq t < T_m/2$ an den Photo-Gates PG_a und PG_b die Potentiale $u_{ma}=U_0+u_m$ und $u_{mb}=0$ sowie anschließend im Zeitbereich $T_m/2 \leq t < T_m$ an den Photo-Gates PG_a und PG_b die Potentiale $u_{ma}=0$ und $u_{mb}=U_0-u_m$ an.

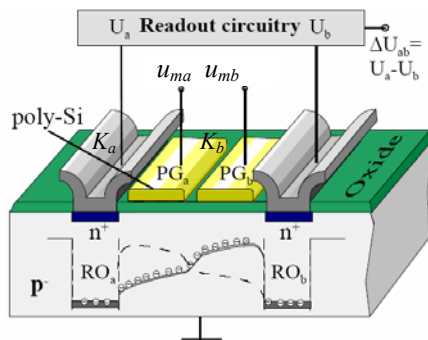


Abb. 2.5: Prinzipieller Aufbau eines PMD-Pixels [55] [56] [65] [67] - [70] [72].

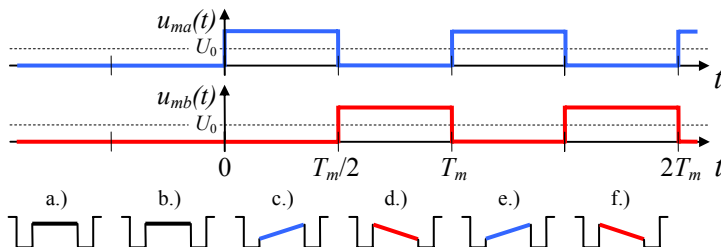


Abb. 2.6: Verschiedene Potentialverläufe, die sog. „Ladungsschaukel“, innerhalb des Halbleitersubstrats eines PMD-Pixels durch Gegentaktmodulation mit u_{ma} und u_{mb} .

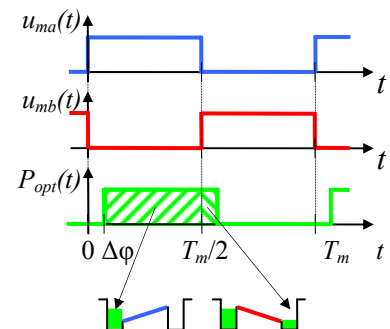


Abb. 2.7: Mischprozess innerhalb des PMD-Pixels.

Das durch die beiden Photo-Gates einfallende Lichtsignal P_{opt} erzeugt nun Ladungstrennungen innerhalb des p-n-dotierten Halbleitersubstrats, wobei die photogenerierten Ladungsträger, je nach Phasenlage $\Delta\varphi$ zwischen u_{ma} bzw. u_{mb} und P_{opt} , dementsprechend zu den beiden Auslekathoden K_a und K_b abfließen (s. **Abb. 2.7**). Die hier dargestellten unterschiedlichen Potentialverläufe und das Sammeln der photogenerierten Ladungsträger innerhalb des Halbleitersubstrats eines PMD-Pixels werden in Schwarte et al. [55] [56] [65] [67] - [70] [72] entsprechend ihrer Arbeitsweise auch „Ladungsschaukel“ genannt.

Das Sammeln von photogenerierten Ladungsträgern geschieht über eine gewisse Zeitspanne, die sog. Integrationszeit T_{int} , hinweg und verkörpert den o.g. Integrator. Die photogenerierten Ladungsträger in K_a und K_b , werden also über die Zeit T_{int} jeweils aufintegriert, verstärkt und als Spannungen U_a und U_b ausgegeben. Die Differenz von U_a und U_b stellt den Korrelationswert dar und entspricht der Phasenlage $\Delta\varphi$ zwischen Sendesignal u_m und moduliertem Empfangssignal $P_{opt}(t)$ [55] [56] [65] [67] - [70] [72]:

$$U_{\Delta} = U_a - U_b \quad (2.10)$$

Eine Summenbildung von U_a und U_b ergibt die Lichtintensität bzw. den Grauwert des entsprechenden Raumpunktes [55] [56] [65] [67] - [70] [72]:

$$U_{\Sigma} = U_a + U_b \quad (2.11)$$

Des Weiteren zeigen die o.g. Gleichungen (2.5a) und (2.5b) auch, dass die Integrationszeit T_{int} stets ein Vielfaches der Periodendauer des modulierten Sendesignals T_m sein darf. D.h. innerhalb des PMD-Pixels wird der Misch- und Integrationsprozess (Korrelationsprozess) n mal durchgeführt, anschließend gemittelt und schließlich als Distanz-, Grau- und Amplitudenwert ausgegeben. Letzterer gibt die Genauigkeit bzw. die Zuverlässigkeit von Distanz- und Grauwert an. Dies hat zur Folge, dass einerseits die Schwankungen bzw. das Videobildrauschen pro PMD-Pixel sich reduzieren, jedoch andererseits sich damit die Frame-Rate verringert.

Zur Rauschunterdrückung von PMD-Videobildern wird lediglich die Integrationszeit T_{int} der PMD-Pixel erhöht. Dadurch können innerhalb dieser Zeit vermehrt photogenerierten Ladungsträger im o.g. Integrator gesammelt werden, wodurch die

Phasenverschiebung $\Delta\varphi$ und die entsprechende Distanz d genauer errechnet werden. Als weitere Maßnahmen erfolgt eine Median-Filterung des gesamten PMD-Videobildes. Damit werden statistische Ausreißer getilgt, wobei Ecken und Kanten weitestgehend unverfälscht bleiben.

2.3.2 Eindeutigkeitsbereich

Gleichung (2.2) $d = \tau \cdot c/2$ beschreibt, dass die Lichtlaufzeit τ stets die entsprechende Distanz d wiedergibt. Bei einer Distanzmessung auf der Basis einer konventionellen Zeitmessung können bei großen Distanzen d auch hinreichend genaue Lichtlaufzeiten τ ermittelt werden. Lediglich die Sendeleistung und das Detektieren schwacher Empfangssignale schränken diese Methode ein. Insbesondere bei weit entfernten Objekten gelangen nur noch schwache Reflektionssignale an den Empfänger, die auch unterhalb des Rauschpegels liegen und mit der Laufzeitmessmethode somit nicht mehr ausgewertet werden können. Des Weiteren fordert die Laufzeitmessung bei kleinen Lichtlaufzeiten τ bzw. kleinen Distanzen d hoch präzise und schnelle Messelektronik, wodurch diese Methode ihre Grenzen findet.

Gegenüber der konventionellen Zeitmessung arbeitet die Phasenmessmethode innerhalb des PMD-Pixels als Korrelationsempfänger. Dies verlangt jedoch, dass das Sendelicht mit einem Modulationssignal überlagert wird. Mit Hilfe der *kkf*-Methode können somit Signale erfasst werden, die bereits unterhalb des Rauschpegels liegen. Darüber hinaus können damit auch kleinste Phasendifferenzen $\Delta\varphi$ und damit kleinste Distanzen d erfasst werden.

Nachteile durch die Verwendung eines modulierten Sendesignals und Auswertung mittels der *kkf*-Methode ergeben sich aufgrund der Periodizität des Sende- bzw. Empfangssignals und damit der Mehrdeutigkeiten der Phasendifferenz $\Delta\varphi$. Die Phasendifferenz $\Delta\varphi$ kann hierbei nur im Bereich $0 \leq \Delta\varphi < 2\pi$ eindeutig bestimmt werden. D.h. bei einer Modulationsfrequenz $f_m = 20$ MHz, $c \approx 300.000$ km/s und $\Delta\varphi_{max} = 2\pi$ ergibt sich ein Eindeutigkeitsbereich der entsprechenden Distanz l_{PMD} von 7,5 m. Laut Gleichung (2.9) kann jedoch die Modulationsfrequenz f_m variiert und somit die entsprechende Distanz l_{PMD} geändert werden. Eine Verringerung der Modulationsfrequenz f_m ergibt eine Vergrößerung der max. eindeutigen Distanz l_{PMD} .

2.4 Vorverarbeitung und Filterung von PMD-Videobilder

Tiefenwertbilder der PMD-Kamera des Typs „PMD [Vision] 3k-S“ weisen hohe Rauschanteile auf (s. **Abb. 2.8** l.), die für die weiteren Bildverarbeitungsschritte herausgefiltert werden müssen. Bei der Rauschunterdrückung bzw. Glättung solcher Videobilder unterscheidet man zwischen linearen und nichtlinearen Glättungsfiltren. Zu den linearen Filtren gehören der Rechteck-, der Binominal-, der Gaußfilter, usw.

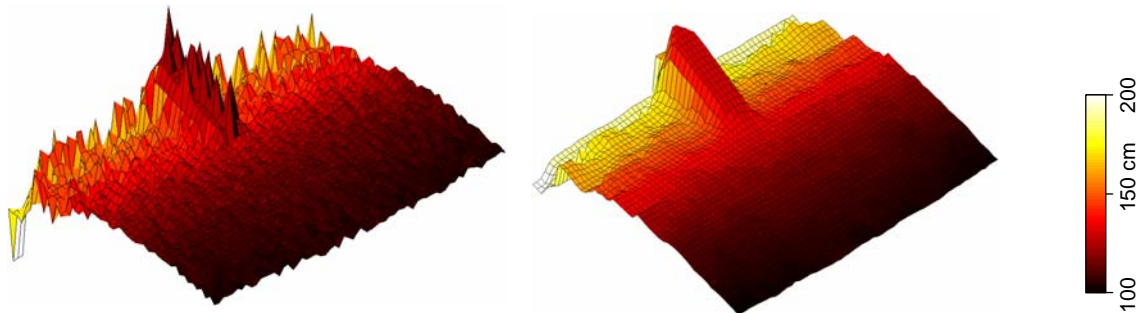


Abb. 2.8: Rauschunterdrückung bei PMD-Videobildern: PMD-Videobild in Original (l.) und mit Integrationszeit-Erhözung und Median-Filterung (r.).

Ein nichtlinearer Filter ist z.B. der klassische Medianfilter, der gewichtete und auch gesteuerte Mittelung ermöglicht. Lineare Filter eignen sich zwar gut zur Unterdrückung von mittelwertfreiem, normalverteiltem Rauschen, da sie insbesondere die hohen Ortsfrequenzen im Videobild unterdrücken, haben aber den Nachteil, dass sie zur Verwischung der Kanten beitragen. Aus diesem Grund wird zur Rauschunterdrückung von PMD-Videobildern der nichtlineare Medianfilter eingesetzt.

Der Medianfilter stellt einen sog. Rangordnungsfiler dar. Hierbei werden pro Pixel zusammen mit seinen acht Umgebungspixeln deren Pixelwerte der Größe nach sortiert. Aus den neun Pixelwerten wird nun der mittlere Pixelwert gewählt und als Filterergebnis dem Zielpixel zugewiesen. Dies hat den Vorteil, dass so neben normalverteiltem Rauschen auch binäres Rauschen entfernt wird. Des Weiteren kommt es bei einem Medianfilter nicht zu der ungewollten Kantenglättung, wie es bei den linearen Filtren der Fall ist [76].

2.5 2D- und 3D-Videobilder und deren Kombination

Grundsätzlich können die PMD-Kamera und eine 2D-Kamera innerhalb des mobilen Fahrzeugs jeweils separat und unabhängig voneinander eingesetzt werden. Die

PMD-Kamera wird dabei für die Hindernis- und Fahrwegserkennung, sowie zur Kartengenerierung der Fahrzeugumgebung verwendet. Die 2D-Kamera dient hauptsächlich der visuellen Kontrolle für den Operator. Darüber hinaus kann sie auch zur Detektierung und Vermessung von künstlicher Referenzmarken verwendet werden (s. Kap. 6.3).

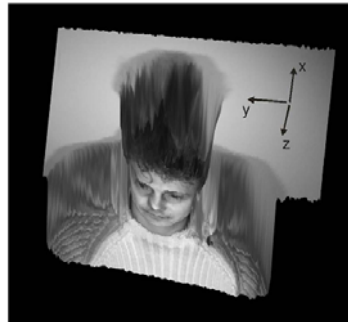


Abb. 2.9: Person als 3D-Tiefenwertbild kombiniert mit 2D-Grauwertbild der PMD-Kamera [75].

Abb. 2.10: Stereovideobild, welches aus einem PMD-Videobild generiert wurde [23].

Eine Kombination von PMD-Tiefenwertbild mit PMD-Grauwertbild oder mit 2D-Videofarbbild kann die Qualität und Verarbeitungsgeschwindigkeit der Anwendungen verbessern. In Driewer [77] wird insbesondere die 2D/3D-Umgebungsvisualisierung von mobilen Fahrzeugen als effektives Werkzeug für den Einsatz in Telemetrie, Telemanipulation und Teleoperation anschaulich beschrieben. Verschiedene Kombinationen von 2D/3D-Videobildern werden beispielsweise in **Abb. 2.9** bis **2.11** gezeigt, die für diesen Einsatz geeignet sind. Insbesondere in **Abb. 2.10** wurden 3D-Tiefenwerte eines PMD-Videobildes in ein Stereovideobild mit Rot- und Grün-Informationen umgewandelt, so dass mit Hilfe einer 3D-Rot-Grün-Brille eine 3D-Ansicht betrachtet werden kann [23].

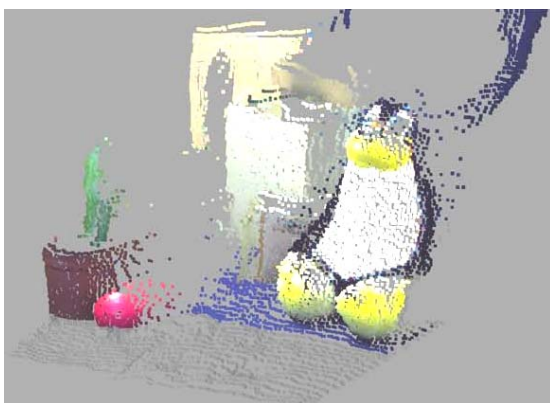


Abb. 2.11: Szene mit „Pinguin“ als 3D-Tiefenwertbild kombiniert mit 2D-Farbwertbild einer 2D-Kamera [46].

Entsprechend **Abb. 2.12.a** kann eine solche Kombination aus PMD-Tiefenwertbild und 2D-Videofarbbild mit Hilfe einer sog. 2D/3D-Parallelkamera realisiert werden. Hier sind eine PMD-Kamera und eine 2D-Kamera exakt parallel zu ihren optischen Achsen im definierten Abstand zueinander ausgerichtet (s. a. **Abb. 3.2** und **4.1**). Die jeweiligen Videobilder werden per Software pixelentsprechend zugeordnet (3D-2D-Registrierung) [36] [46] [78] [79] [80].

Eine monokulare 2D/3D-Kombikamera, die sog. 2D/3D-Multichip-Kamera (s. **Abb. 2.2.a** und **Abb. 2.12.b**), vereinigt alle Vorzüge einer 2D-CMOS-Kamera und der PMD-Kamera durch komplementäre Nutzung des Modulations- und Fremdlichtes über eine gemeinsame Optik und durch pixelsynchrone 3D- und 2D-Bildverarbeitung. Sie besteht aus einem RGB-CMOS-Chip und einem PMD-Chip, die über einen spektralen Strahlteiler [39] [40] optisch synchron gekoppelt sind. Über den spektral selektiven Strahlteiler wird das reflektierte modulierte Sendelicht der 3D-Szene mit einer z.B. 19.200er PMD-Matrix als 3D-Tiefenwertbild und als 2D-Grauwertbild aufgenommen. Das Restlicht, im Allgemeinen das Fremdlicht, passiert den Strahlteiler und wird als 2D-Farbwertbild mit einer lateral ca. 25-100-fach höher auflösenden CMOS-Farbkamera pixelsynchron kombiniert aufgenommen. Insgesamt entstehen also drei Videobilder: ein 3D-Tiefenwertbild, ein 2D-Grauwertbild und ein 2D-Farbwertbild.

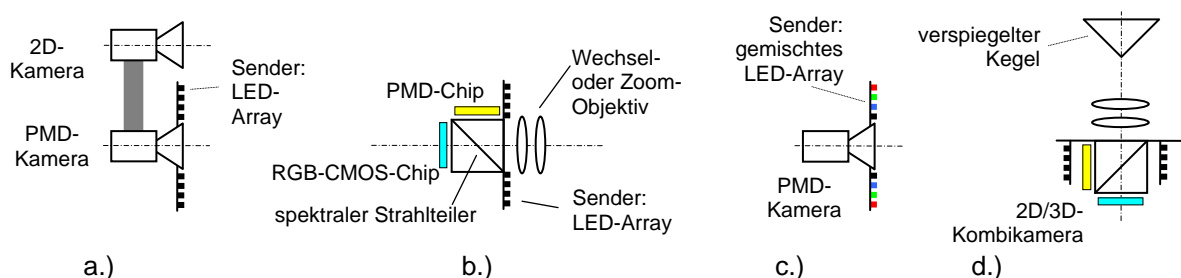


Abb. 2.12: Prinzipielle Darstellungen einer a.) 2D/3D-Parallelkamera, b.) 2D/3D-Multichip-Kamera (s. a. **Abb. 2.2.a**) [35], c.) PMD-Farbkamera und d.) 2D/3D-Omni-CombiCam [81] [82] [83] [84].

Bei beiden o.g. Systemen, 2D/3D-Parallelkamera (**Abb. 2.12.a**) und 2D/3D-Multichip-Kamera (**Abb. 2.12.b**), besteht die Problematik, dass hier unterschiedliche Videobildgrößen und -auflösungen miteinander kombiniert werden müssen, z.B. 64x48, 160x120 oder 204x204 PMD-Pixel mit z.B. 1280x1024 CCD-Pixel. Bei der 2D/3D-Parallelkamera muss zusätzlich, ähnlich einem Stereovision-System, der fixe Abstand beider Kameras zueinander bedacht werden.

Nach einer Idee von R. Schwarte [81] - [84] könnte eine sog. PMD-Farbkamera (**Abb. 2.12.c**) so aufgebaut werden, dass die Sendeeinheit einer PMD-Kamera statt aus IR-LEDs aus Rot-, Grün- und Blau-LEDs besteht. Die jeweiligen LED-Farbgruppen würden im Zeitmultiplex abwechselnd moduliertes rotes, grünes oder blaues Licht senden und der entsprechend synchronisierte PMD-Chip könnte aus den reflektierten Lichtsignalen neben dem 3D-Tiefenwertbild jetzt auch ein 2D-Rotwertbild, ein 2D-Grünwertbild und ein 2D-Blauwertbild generieren. Aus diesen 2D-Videobildern lässt sich ein 2D-RGB-Videofarbbild erzeugen.

In vielen Navigations- und Explorationsaufgaben sowie diversen Robotik-anwendungen besteht der Wunsch nach räumlicher Rundumsicht. In [81] - [84] wurde daher der Aufbau einer 2D/3D-Omni-CombiCam (**Abb. 2.12.d**) angeregt, die, ähnlich einer 2D-Omnidirektionalkamera (s. Kap. 4.2.6), mit einem kegelförmigen Spiegel ausgestattet ist. Die Beleuchtungseinheit muss allerdings so angeordnet sein, dass kein Sendelicht direkt über den kegelförmigen Spiegel empfangen wird. Eine andere Lösungsmöglichkeit bietet eine bewegliche PMD-Kamera, die auf einem um 360° drehbaren Adapter fixiert ist. Denkbar sind auch 4 oder 6 sternförmig angeordnete Weitwinkel-PMD-Kameras mit jeweils 90°- oder 60°-Weitwinkel-Objektiv.

3 Hindernis- und Fahrwegerkennung mit der PMD-Kamera

Der Frontbereich des mobilen Fahrzeugs wird in einen linken, einen mittlinken, einen mitterechten und einen rechten Sensorbereich, im Weiteren Segmente genannt, eingeteilt. Diese Unterteilung in vier Segmente soll einheitlich für alle hier verwendeten Hinderniserkennungssysteme einschließlich der PMD-Kamera gelten. Damit können mehrere Hinderniserkennungssysteme parallel arbeiten und benötigen lediglich ein einziges in Kap. 5 beschriebenes Hindernisvermeidungssystem. Sowohl diese Vereinheitlichung als auch die hier gewählte Anzahl von vier Segmenten reduziert letztendlich den Hard- und Software-Aufwand der Hindernisvermeidung, d.h. bei vier Segmenten bleibt die Exaktheit der Hindernisvermeidung noch hinreichend gegeben und dies bei einer relativ kleinen Anzahl von Sensoren pro Hinderniserkennungssystem. Auch geht die Segment-Nummer als Variable in das Hindernisvermeidungssystem ein und beeinflusst mit dessen Wertebereich, hier 0 bis 3, die Echtzeitfähigkeit der implementierten Algorithmen.

Hindernisse mit genügender Höhe und Breite innerhalb der Fahrzeugumgebung können mit Hilfe des mechanischen Front-Bumpers und des IR-Front-Bumpers ausreichend gut erfasst werden. Auch negative Hindernisse (s. Kap. 3.4) mit genügender Ausdehnung sind mit dem IR-Boden-Bumper auffindbar. Hindernisse, die jedoch ober- oder unterhalb des Erfassungsbereichs eines mechanischen Sensors oder IR-Sensors liegen, können nicht detektiert werden. Ebenso problematisch ist es, wenn sich das Hindernis zwischen den Erfassungsbereichen zweier benachbarter IR-Sensoren befindet (s. Kap. 4.2.2).

Diese Probleme können durch den Einsatz einer PMD-Kamera vermieden werden. Da die PMD-Kamera den Frontbereich des mobilen Fahrzeugs als gesamten 3D-Raum ohne Lücken erfasst, können auch sehr dünne bzw. schmale Hindernisse detektiert werden. Voraussetzung dafür ist jedoch, dass der PMD-Chip eine genügend hohe Pixel-Auflösung besitzt.

3.1 Integration der PMD-Kamera in ein mobiles Fahrzeug

Die PMD-Kamera besitzt bei einer Modulationsfrequenz des IR-Lichtes von $f_m = 20$ MHz einen Eindeutigkeitsbereich der Distanzwerte l_{PMD} von 7,5 m (s. Kap. 2.3). Beim

Einbau der PMD-Kamera in ein mobiles Fahrzeug muss daher gewährleistet sein, dass diese keine Reflexionen von Objekten außerhalb ihres Eindeutigkeitsbereichs empfängt. Die PMD-Kamera muss folglich mit einem bestimmten Neigungswinkel φ_{PMD} eingebaut werden (**Abb. 3.1**).

Des Weiteren sollen mit Hilfe der PMD-Kamera sowohl positive als auch sog. negative Hindernisse erfasst werden (s. Kap. 3.4). Dazu muss diese mit einer bestimmten Montagehöhe h_K auf den Frontbereich des mobilen Fahrzeugs schauen können. Darüber hinaus unterliegt der Neigungswinkel φ_{PMD} zwei weiteren Einschränkungen: Bei hoher Fahrzeuggeschwindigkeit sollte der ferne Fahrweg bei der Distanz $d \approx d_{max}$ möglichst weit vorausschauend einsehbar sein, d.h. $\varphi_{PMD} > 0$. Dagegen ist es notwendig, dass bei niedriger Fahrzeuggeschwindigkeit der nahe Frontbereich $d > d_{min}$ vor dem mobilen Fahrzeug erfasst werden kann, d.h. $\varphi_{PMD} < 90^\circ$.

Insgesamt müssen bei der Wahl des optimalen Neigungswinkels, im Bereich $0^\circ < \varphi_{PMD} < 90^\circ$, der maximale Eindeutigkeitsbereich l_{PMD} , der optische Öffnungswinkel $\varphi_{Opt.}$ des Kameraobjektivs und die Montagehöhe h_K berücksichtigt werden (s. **Abb. 3.1**). Der Einfluss der Fahrgeschwindigkeit des mobilen Fahrzeugs mit $v_F < 1 \text{ m/s}$ soll hier vernachlässigt werden.

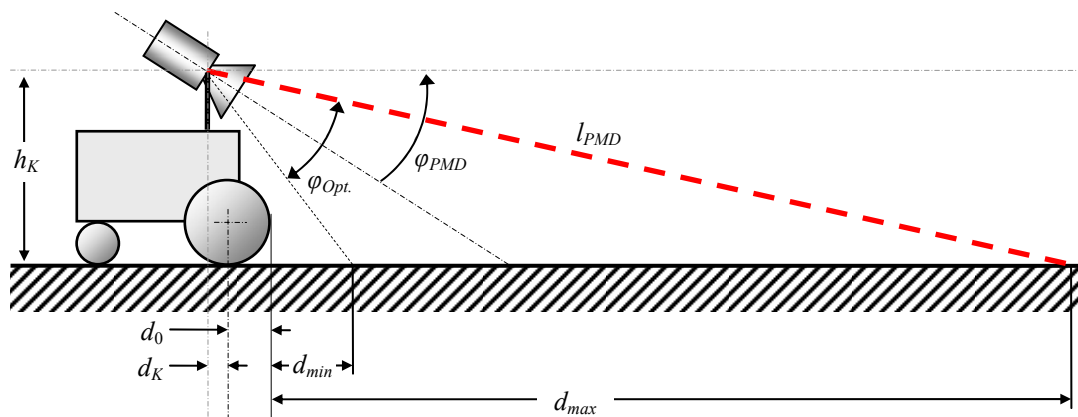


Abb. 3.1: Neigungswinkel φ_{PMD} der PMD-Kamera bei der Integration in ein mobiles Fahrzeug.

Anhand der in **Abb. 3.1** gezeigten geometrischen Betrachtung können folgende drei Beziehungen (3.1) (3.2) und (3.3) abgeleitet werden. Hierbei stellen die Konstanten d_0 , d_K und h_K Distanzen dar, die direkt am mechanischen System messtechnisch zu ermitteln sind. Der optische Öffnungswinkel $\varphi_{Opt.}$ ist dem Datenblatt der PMD-Kamera

zu entnehmen und der Eindeutigkeitsbereich entspricht $l_{PMD} = 7,5$ m. Aus diesen Beziehungen lässt sich schließlich der Neigungswinkel φ_{PMD} errechnen:

$$\text{Neigungswinkel } \varphi_{PMD} : \quad \sin\left(\left|\varphi_{PMD}\right| - \frac{\varphi_{Opt.}}{2}\right) = \frac{h_K}{l_{PMD}} \quad (3.1)$$

$$\text{Minimale Distanz } d_{min} : \quad \tan\left(\left|\varphi_{PMD}\right| + \frac{\varphi_{Opt.}}{2}\right) = \frac{h_K}{d_K + d_0 + d_{min}} \quad (3.2)$$

$$\text{Maximale Distanz } d_{max} : \quad l_{PMD}^2 = (d_K + d_0 + d_{max})^2 + h_K^2 \quad (3.3)$$

Die PMD-Kamera und ggf. auch eine 2D-Kamera können nun mit diesem Neigungswinkel φ_{PMD} mit Hilfe des in **Abb. 3.2** dargestellten Adapters auf dem mobilen Fahrzeug fixiert werden, so dass der Eindeutigkeitsbereich nicht überschritten wird. Um den Eindeutigkeitsbereich zu bewahren, darf die längste Sichtlinie der PMD-Kamera den Distanzwert $l_{PMD} = 7,5$ m nicht überragen.

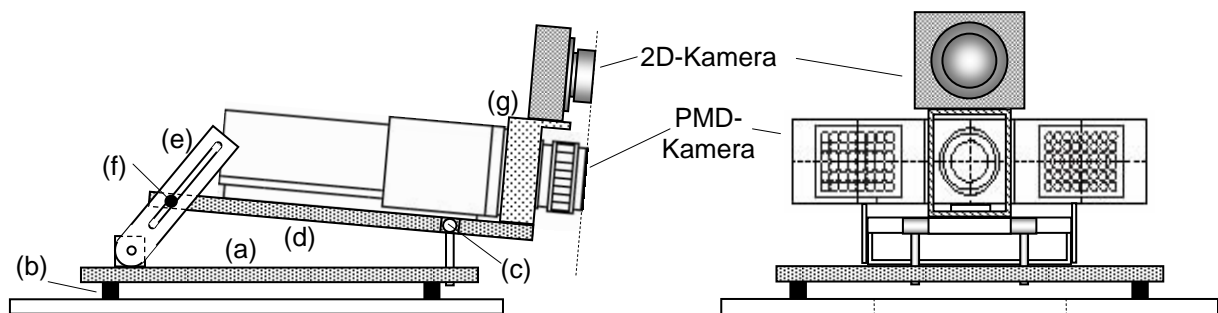


Abb. 3.2: Mechanische Vorrichtung eines Adapters zur Integration der PMD-Kamera und ggf. der 2D-Kamera in ein mobiles Fahrzeug.

Der in **Abb. 3.2** dargestellte Adapter besteht aus einer auf vier mechanischen Schwingungsdämpfern (b) montierten Basisplatte (a) aus Plexiglas und einer in Punkt (c) drehbar angebrachten Kameraplatte (d) aus Aluminium, die mittels einer beweglichen Führungsleiste (e) im gewünschten Neigungswinkel fixiert (f) werden kann. Während die PMD-Kamera direkt auf die Kameraplatte (d) montiert wird, muss die 2D-Kamera mit Hilfe des mechanischen Bauteils (g) angefügt werden.

3.2 Ausnahmesituationen

Aufgrund des voreingestellten Neigungswinkels der PMD-Kamera mit $0^\circ < \varphi_{PMD} < 90^\circ$ kann diese nicht den gesamten Frontbereich des mobilen Fahrzeugs komplett aufnehmen. Stattdessen verbleibt zwischen maximaler geometrischer Front-

ausdehnung d_0 des mobilen Fahrzeugs und dem Sichtbereich $d_{max} - d_{min}$ der PMD-Kamera ein Blindbereich d_{min} (s. **Abb. 3.1**). Dieser Bereich soll mit Hilfe anderer Hinderniserkennungssysteme, wie dem mechanischen Front-Bumper, dem IR-Front-Bumper und dem IR-Boden-Bumper, überwacht werden (s. Kap. 4.2.1 und 4.2.2).

Des Weiteren kann bei der Fahrt in einem unebenen 2D-Areal das mobile Fahrzeug über einen flachen Gegenstand, der nicht als Hindernis erkannt wurde, hinweg fahren, wie z.B. einen Absatz, eine dünne Kabelleitung, usw. Auch können minimale Unebenheiten und Vertiefungen im Fahrweg des mobilen Fahrzeugs zu Situationen führen, wie sie in **Abb. 3.3** exemplarisch dargestellt sind.

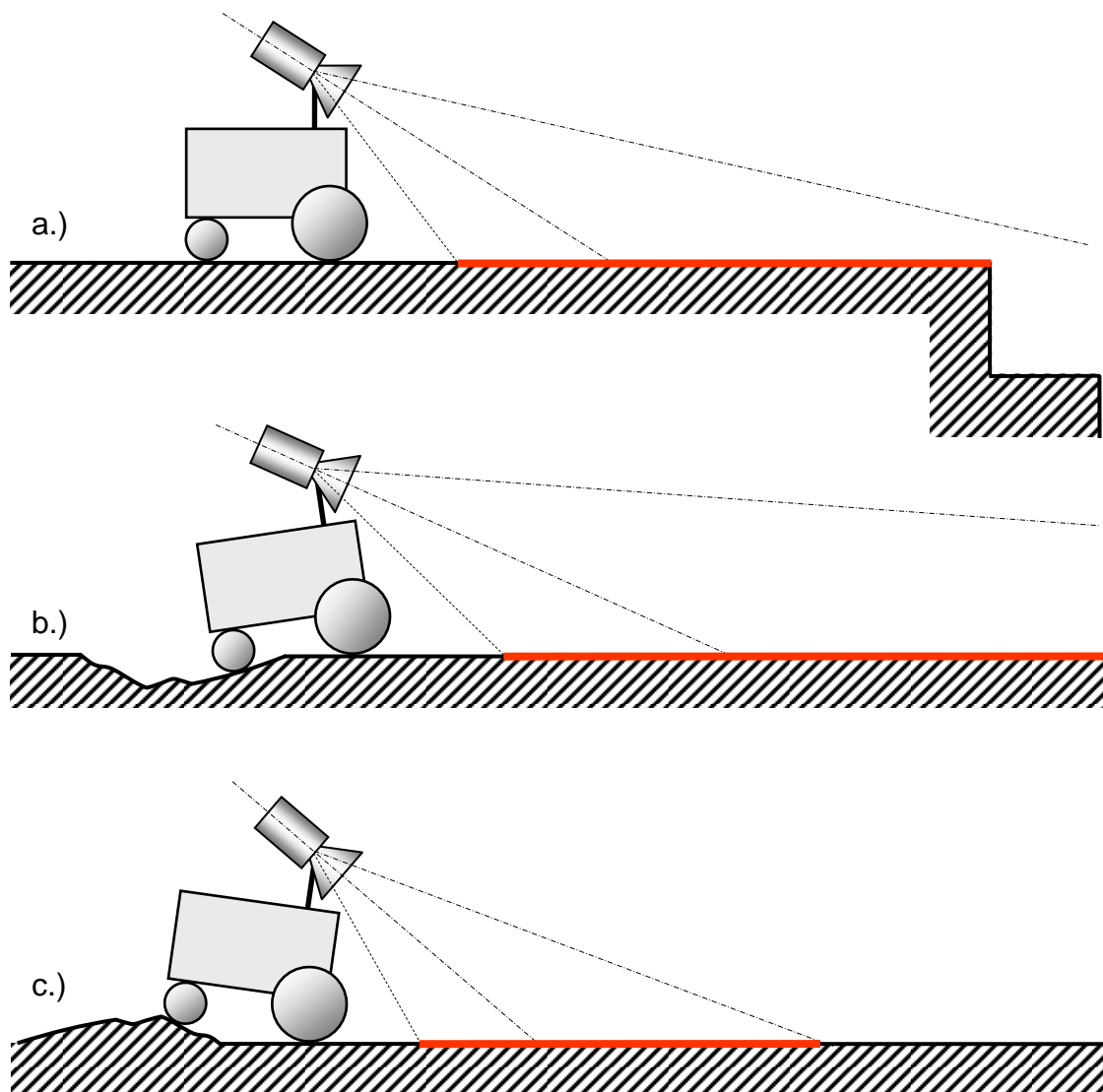


Abb. 3.3: Ausnahmesituationen innerhalb eines quasi flachen 2D-Areals. U.a. beim a.) Anfahren einer hinabführenden Treppe, b.) Durchfahren einer Vertiefung und c.) Hinunterfahren von einer Erhöhung.

So erscheinen beim Heranfahren des mobilen Fahrzeugs an eine hinabführende Treppe im oberen Bereich des entsprechenden PMD-Videobildes mehrdeutige Tiefenwerte aufgrund der Überschreitung des Eindeutigkeitsbereichs mit $d > 7,5$ m (**Abb. 3.3.a** und **3.4.c**). Mehrdeutige Tiefenwerte können mit Hilfe der dazugehörigen Amplitudenwerte ausfindig gemacht werden. Der jeweilige Amplitudenwert kann neben dem Tiefenwert zusätzlich aus der PMD-Kamera pro Bildpixel gelesen werden. Entsprechend Kap. 2.3 und Gleichung (2.7) wird der Amplitudenwert aus der Kreuzkorrelationsfunktion von Sendesignal und Empfangssignal pro Bildpixel errechnet. Ein hoher Amplitudenwert eines Bildpixels gibt eine hohe Genauigkeit des entsprechenden Distanzwertes an und umgekehrt. So können Distanzwerte mit geringen Amplitudenwerten für die weitere Bildverarbeitung ausgesondert werden.

Beim Durchfahren einer Vertiefung bzw. beim Hinunterfahren von einer Erhöhung in **Abb. 3.3.b** und **3.3.c** verändert sich der Neigungswinkel der PMD-Kamera relativ zum Fahrweg. Diese Winkeländerung entspricht dem sog. Pitch-Winkel. Der Pitch-Winkel φ_{Pitch} , der Roll-Winkel φ_{Roll} und der Yaw-Winkel φ_{Yaw} können mit Hilfe eines Inklinationssensor gemessen oder entsprechend Kap. 6.8 aus der Folge zweier PMD-Videobilder berechnet werden. Die Einbeziehung von Roll- und Pitch-Winkel bei der Hindernis- und Fahrwegerkennung markiert hier nun den Übergang von Fahrzeugbewegungen im 2D- hin zu 3D-Arealen. Der Yaw-Winkel entspricht hierbei dem Orientierungswinkel α bzw. $\Delta\alpha$ des mobilen Fahrzeugs.

Anstatt die PMD-Kamera mit Hilfe des in **Abb. 3.2** gezeigten Adapters starr auf dem mobilen Fahrzeug zu fixieren, könnte diese auch auf eine sog. Schwenk-Neige-Einheit (Pan-Tilt-Unit) [85] montiert und am mobilen Fahrzeug befestigt werden. Dies hätte den Vorteil, dass die z.B. mit dem Inklinationssensor gemessenen Roll- und Pitch-Winkel kompensiert werden und die PMD-Kamera stets mit dem gewünschten Neigungswinkel φ_{PMD} ausgerichtet bleibt, auch wenn sich das mobile Fahrzeug durch ein unebenes Areal entsprechend **Abb. 3.3.b** und **3.3.c** bewegt.

3.3 Varianten von Fahrwegen

In den **Abb. 3.4** und **3.5** sind verschiedene Fahrwege ohne Hindernisse im Frontbereich eines mobilen Fahrzeugs dargestellt, die sowohl mit einer 2D- als auch

mit der PMD-Kamera aufgenommen wurden. Beide Kameras sind zusammen auf den in **Abb. 3.2** gezeigten Adapter mit einem Neigungswinkel $\varphi_{PMD} \approx 35^\circ$ und einer Montagehöhe $h_K \approx 40$ cm auf das mobile Fahrzeug montiert.

In **Abb. 3.4.a** wird ein glatt gebohrter Fußboden gezeigt, der aus Kunststoffplatten von ca. 50×50 cm besteht, die an den Kanten mit speziellen Kunststoffstreifen (Breite ≈ 5 mm) aneinander verklebt wurden. Im PMD-Videobild zeigen sich beide Materialien durch unterschiedliches Reflexionsverhalten des IR-Lichtes der PMD-Kamera. Insbesondere die Kunststoffplatten verfälschen aufgrund der Totalreflexion die Distanzwerte im oberen bzw. fernerem PMD-Videobildbereich.

Zwei Phänomene können festgestellt werden: 1.) Ähnlich einer „Fata Morgana“ [86] bildet der Boden die Reflexionsschicht. So strahlt das IR-Licht vom PMD-Sender über die Totalreflexion am Boden an ein fernes Objekt, dort erfolgt eine Reflexion und über eine erneute Totalreflexion am Boden gelangt das IR-Licht schließlich zurück an das Bildpixel. 2.) Auf der Wegstrecke PMD-Sender, Boden, fernes Objekt, Boden, Bildpixel wird das IR-Licht über die maximale Distanz des Eindeutigkeitsbereichs l_{PMD} hinausgeführt und erzeugt so Mehrdeutigkeiten innerhalb der Bildpixel (s. Kap. 2.3.2).

Ein nahezu ideales PMD-Videobild ist in **Abb. 3.4.b** zu sehen. Hier wurde eine weiße Papierbahn als Fahrweg aufgenommen. Das Reflexionsverhalten dieser matten Oberfläche ähnelt einer Lambertschen Fläche [87] [88] und erzeugt daher keine Verfälschungen im reflektierten IR-Licht der PMD-Kamera. Des Weiteren ist in **Abb. 3.4.b** klar zu erkennen, dass die PMD-Kamera die ebene Fahrwegfläche als gewölbte Fläche im Tiefenwertbild wiedergibt. An dieser Stelle wird deutlich, dass die PMD-Kamera die Umgebung generell in Kugelkoordinaten aufnimmt und daher für die weitere Bildverarbeitung stets eine Konvertierung des Tiefenwertbildes von Kugelkoordinaten in kartesische Koordinaten erfolgen muss.

Was in **Abb. 3.4.a** bereits angedeutet ist, wird in **Abb. 3.4.c** in extremer Weise veranschaulicht. Hier wurde eine hinabführende Treppe an einem Fahrweg aufgenommen. Oberhalb der ersten Stufenkante reicht das IR-Licht der PMD-Kamera über die maximale Distanz des Eindeutigkeitsbereichs l_{PMD} hinaus und

erzeugt Mehrdeutigkeiten in den entsprechenden Bildpixeln. Im PMD-Videobild sind diese Mehrdeutigkeitswerte in Form eines Phantomhindernisses zu sehen (oberer Bereich des PMD-Videobildes in **Abb. 3.4.c**, u.). Zur Unterdrückung solcher Erscheinungen hilft auch hier die Auswertung des Amplitudenwertes pro Bildpixel, die bei kleinen Werten ausgesondert werden können.

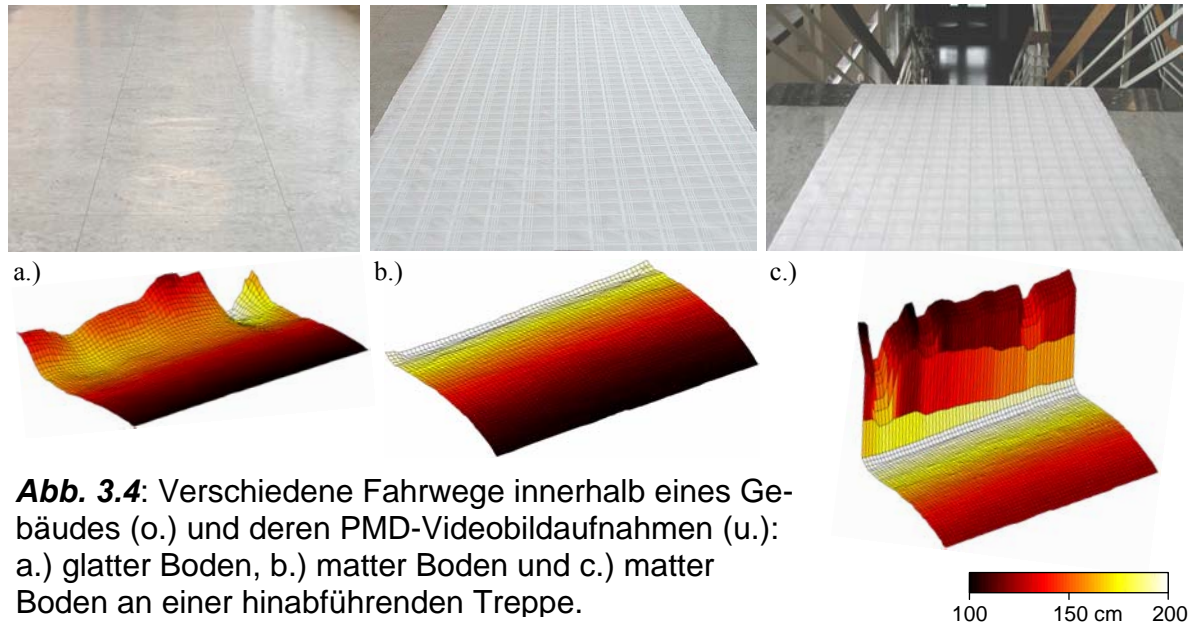


Abb. 3.4: Verschiedene Fahrwege innerhalb eines Gebäudes (o.) und deren PMD-Videobildaufnahmen (u.): a.) glatter Boden, b.) matter Boden und c.) matter Boden an einer hinabführenden Treppe.

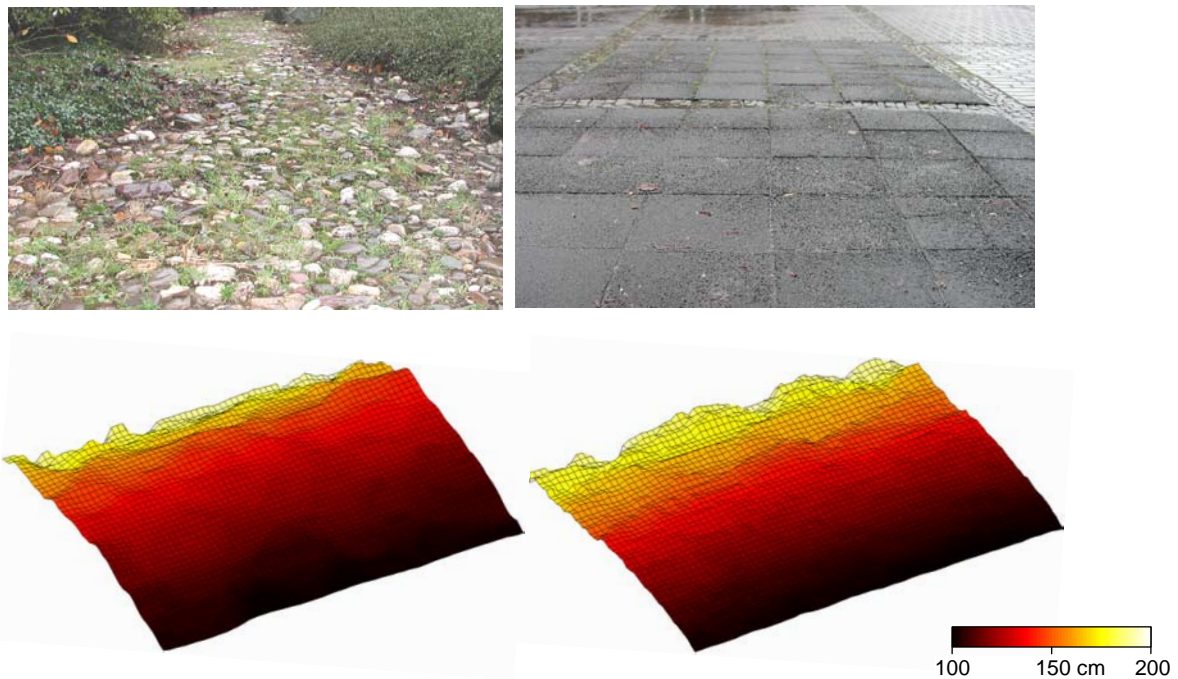


Abb. 3.5: Fahrwege im Außenbereich und deren PMD-Videobildaufnahmen: Steiniger Gartenweg (l.) und nasse Steinplatten (r.).

Die in **Abb. 3.5** dargestellten Aufnahmen veranschaulichen, dass die PMD-Kamera für Außenaufnahmen gut geeignet ist. Der in **Abb. 3.5** l. zu sehende Gartenweg mit grob strukturierter Oberfläche wird im PMD-Videobild mit geringen Störungen abgebildet. Selbst der in **Abb. 3.5** r. gezeigte Fahrweg mit nassen Steinplatten, weist keine Totalreflexionen auf. Es muss jedoch angenommen werden, dass zwar Totalreflexionen existieren, aber hier kein fernes Objekt vorhanden war, welches das IR-Licht hätte reflektieren können.

3.4 Hindernis- und Fahrwegerkennung

In Schilling/Roth [89] werden Hindernisse in sog. positive und negative Hindernisse differenziert. Demnach sind positive Hindernisse Erhöhungen, Objekte, Gegenstände und auch Personen, die sich auf bzw. über dem Fahrweg befinden. Dagegen stellen negative Hindernisse Vertiefungen, Löcher, Dellen, hinabführende Treppenstufen, usw. innerhalb des Fahrwegs dar. Gemäß Kap. 3.1 wurde die PMD-Kamera mit dem Neigungswinkel φ_{PMD} auf dem mobilen Fahrzeug montiert. Die damit realisierte Blickrichtung der PMD-Kamera auf die Fahrwegszene ermöglicht es, dass sowohl positive und negative Hindernisse als auch der Fahrweg selbst gemeinsam im PMD-Videobild aufgenommen werden (**Abb. 3.6** und **3.7**). Innerhalb der Bildverarbeitung gilt es nun im PMD-Videobild zwischen Hindernissen und Fahrweg zu unterscheiden.

Aufgrund des Neigungswinkels $\varphi_{PMD} \approx 35^\circ$ der PMD-Kamera erscheint die damit aufgenommene Szene innerhalb des PMD-Videobildes, die mit φ_{PMD} um eine horizontale Achse rotiert wurde. Die horizontale Achse entspricht der x -Achse gemäß **Abb. 3.6.c**. Die Bildverarbeitung in der Software „Sensor-Server“ (s. Kap. 4.3.3) kompensiert dies, indem jedes Pixel $\mathbf{d}(x,y,z)$ des PMD-Videobildes (**Abb. 3.6.c** und **3.7.c**) mit Gleichung (3.6) wiederum mit dem Neigungswinkel $-\varphi_{PMD}$ um die x -Achse rotiert wird (**Abb. 3.6.d** und **3.7.d**). Allerdings muss zuvor aus dem Bildpixel $d_{2D}(x,y)$ ein $\mathbf{d}_{3D}(x,y,z)$ mit Gleichung (3.4) erzeugt werden. Des Weiteren ist das Bildkoordinatensystem in **Abb. 3.6.c** und **3.7.c** durch Translation (3.5) mit $\mathbf{t} = (31, 23, 0)^T$ in das Bildzentrum zu transformieren [90] [91].

$$\mathbf{d}_{3D}(x,y,z) = \begin{pmatrix} x \\ y \\ d_{2D}(x,y) \end{pmatrix} \quad \text{mit } x = 0, 1, \dots, 63 \text{ und } y = 0, 1, \dots, 47 \quad (3.4)$$

$$\mathbf{d}(x,y,z) = \mathbf{d}_{3D}(x,y,z) - \mathbf{t} \quad (3.5)$$

$$\mathbf{d}^*(x,y,z) = \mathbf{R}_x(-\varphi_{PMD}) \cdot \mathbf{d}(x,y,z) \quad (3.6)$$

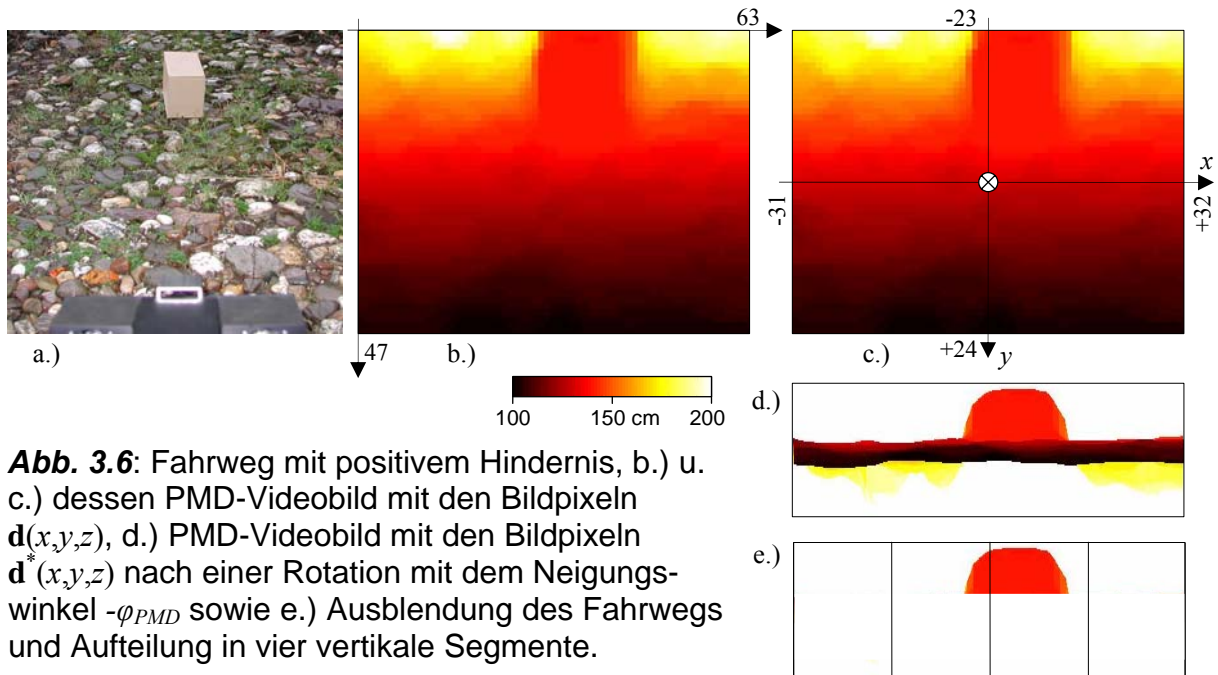


Abb. 3.6: Fahrweg mit positivem Hindernis, b.) u. c.) dessen PMD-Videobild mit den Bildpixeln $\mathbf{d}(x,y,z)$, d.) PMD-Videobild mit den Bildpixeln $\mathbf{d}^*(x,y,z)$ nach einer Rotation mit dem Neigungswinkel $-\varphi_{PMD}$ sowie e.) Ausblendung des Fahrwegs und Aufteilung in vier vertikale Segmente.

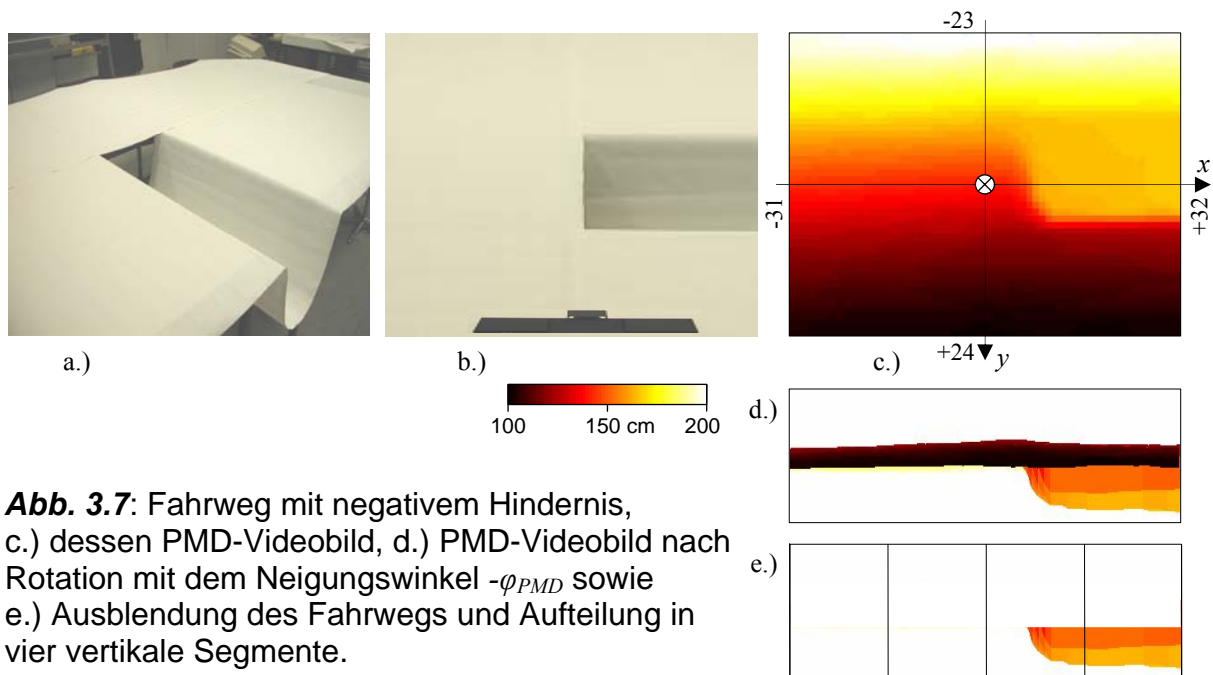


Abb. 3.7: Fahrweg mit negativem Hindernis, c.) dessen PMD-Videobild, d.) PMD-Videobild nach Rotation mit dem Neigungswinkel $-\varphi_{PMD}$ sowie e.) Ausblendung des Fahrwegs und Aufteilung in vier vertikale Segmente.

Im Falle einer Fahrzeugbewegung innerhalb einer 3D-Umgebung können hier die z.B. vom Inklinationssensor (s. Kap. 4.2.2) gemessenen Roll- und Pitch-Winkel in folgender Weise berücksichtigt werden: Nach der Verschiebung der Bildkoordinaten in das Bildzentrum mit (3.5) wird jedes Bildpixel $\mathbf{d}(x,y,z)$ nacheinander mit den

Gleichungen (3.7) (3.8) mit $-\varphi_{Roll}$ um die z -Achse und mit $(-\varphi_{Pitch} - \varphi_{PMD})$ um die x -Achse rotiert (s. Kap. 10.1).

$$\mathbf{d}^* = \mathbf{R}_z(-\varphi_{Roll}) \cdot \mathbf{d} \quad (3.7)$$

$$\mathbf{d}^{**} = \mathbf{R}_x(-\varphi_{Pitch} - \varphi_{PMD}) \cdot \mathbf{d}^* \quad (3.8)$$

Das Ergebnis ist in **Abb. 3.6.d** und **3.7.d** verdeutlicht. Hier wird der Fahrweg jetzt als horizontaler Balken dargestellt und das eigentliche Hindernis befindet sich oberhalb bzw. unterhalb des Balkens. Anhand der Lage des Hindernisses ober- und unterhalb dieses horizontalen Balkens lässt sich ein positives oder ein negatives Hindernis detektieren.

Während der Fahrt eines mobilen Fahrzeugs verändert sich die Position und Gestalt des horizontalen Balkens innerhalb der PMD-Videobilder in **Abb. 3.6.d** und **3.7.d** nur dann nicht, wenn für die ebene 2D-Umgebung die Gleichung (3.6) und für die unebene 3D-Umgebung die Gleichungen (3.7) (3.8) auf das jeweilige Bildpixel angewandt werden.

Als Ergebnis erhält man ein sog. lageneutrales PMD-Videobild. Erst damit können die Bildpixel ausgeblendet werden, die zum horizontalen Balken und damit zum Fahrweg gehören. Hierdurch wird eine Trennung von Fahrweg und Hindernissen vollzogen. Übrig bleiben die Hindernisse (positive und negative Hindernisse), die für die Hindernisvermeidung relevant sind. Diese Separierung kann nun kontinuierlich in jedem nacheinander aufgenommenen PMD-Videobild durchgeführt werden.

Nach der Separierung von Fahrweg und Hindernissen erfolgt nun die eigentliche Hinderniserkennung. Grundsätzlich soll hierbei lediglich das nächstgelegene Hindernis relativ zum mobilen Fahrzeug berücksichtigt werden. Dazu wird das PMD-Videobild, entsprechend **Abb. 3.6.e** und **3.7.e** ohne Fahrweg, in vertikale Segmente aufgeteilt. Gemäß Kap. 4.2.1 und in Hinblick auf die Bewegungssteuerung in Kap. 4.3.5.d wurde eine Aufteilung in vier Segmente mit den Segment-Nummern $N_{Segment} = 0, 1, 2, 3$ gewählt.

Die hier scheinbar willkürlich gewählte Einteilung in vier Segmente berücksichtigt, dass bei höherer Quantisierung in vertikale Bildsegmente die horizontale Hindernisposition zwar genauer bestimmt würde, jedoch der damit verbundene

Rechenaufwand sich erhöht, und umgekehrt. Die Anzahl der Segmente muss entsprechend der Bewegungssteuerung in Kap. 4.3.5.d stets eine gerade Zahl sein, kann minimal zwei und maximal 64 (PMD-Videobildbreite) betragen. Wegen der Echtzeitfähigkeit der Hinderniserkennung und -vermeidung, sollte sie jedoch eher klein sein.

Innerhalb jedes Segments wird die kleinste Distanz d_{min} aus den PMD-Daten ermittelt. Darauf folgt die Bestimmung des Segments mit seiner Segment-Nummer $N_{Segment}$, in welchem die kleinste Distanz liegt. Die kleinste Distanz d_{min} und die entsprechende Segment-Nummer $N_{Segment}$ sind das Resultat der Hinderniserkennung mittels PMD-Kamera. Dieses Daten-Tupel wird nun an die Hindernisvermeidung gegeben und innerhalb zweier Fuzzy-Logic-Regler weiterverarbeitet (s. Kap. 5.2).

4 Die mobilen Roboter Tom3D und Merlin3D

Im Allgemeinen ist die Architektur eines mobilen Fahrzeugs als Sensor-Aktor-Controller-System zu betrachten, welches in Interaktion mit seiner physischen Umwelt, aber auch mit dem Operator steht [67]. Dabei kann der Operator direkt vor Ort sein und seinem mobilen Fahrzeug Anweisungen zur Durchführung einer Dienstleistung geben. In diesem Fall spricht man von einem Assistenzsystem (s. **Abb. 1.2**).

Der Operator kann aber auch von weit entfernt via Telemetrie, z.B. WLAN und Internet, Zugriff auf sein mobiles Fahrzeug haben. Dieser Zugriff geht einher mit variablen Antwort- und Totzeiten [92] - [95]. Um Verzögerungen dieser Art zu überwinden, verlangt ein derart vernetztes mobiles Fahrzeug einen gewissen Grad an autonomem Verhalten. Via Telemetrie soll also ein mobiles Fahrzeug ferngesteuert und teilautonom agieren können. Hierbei handelt es sich um eine sog. Telemaschine oder einen Teleroboter (s. **Abb. 1.3**).

Überdies kann ein mobiles Fahrzeug sich nur auf ebener 2D-Fläche oder zusätzlich im ungeordneten 3D-Areal bewegen. Dabei erfordern ferngesteuerte und teilautonome Aktionen auch, dass die entsprechende Software die mechanischen Eigenschaften des mobilen Fahrzeugs, also die Kinematik und Dynamik, berücksichtigt [96] [97]. Dies betrifft insbesondere den Schwerpunkt, die geometrischen Abmessungen, die Art des Antriebes, usw. So entscheidet beispielsweise die Lage des Schwerpunkts ob das mobile Fahrzeug einen Fahrweg mit bestimmter Steigung befahren darf, ohne dass es dabei kippt. Dies gilt auch für die Breite des mobilen Systems. Sie bestimmt, ob eine Gasse bestimmter Breite für das mobile Fahrzeug einen Fahrweg oder ein Hindernis darstellt.

4.1 Fahrtriebe

Abgesehen vom Gehmechanismus bei Humanoiden und bei Insektenantrieben, die hier nicht weiter berücksichtigt werden, sind die meisten mobilen Fahrzeuge mit einem Radantrieb ausgestattet. Hierbei finden sich insbesondere der Ackermann-, der Differential- und der Fahr- / Lenkantrieb:

- Der Ackermannantrieb (s. **Abb. 4.1** r. und **Abb. 4.2** r.) stellt ein Vierrad-Fahrzeug dar, wobei ein Räderpaar der Lenkung und das andere Räderpaar dem Antrieb dient (Hinterradantrieb). Ggf. können für den Antrieb auch das Lenk-Räderpaar (Vorderradantrieb) oder alle Räder (Allradantrieb) genutzt werden. Bei der Navigation muss hierbei auch der Lenkradius berücksichtigt werden.
- Der Differentialantrieb (s. **Abb. 4.1** l., M. und **Abb. 4.2** l.) besteht aus drei Rädern, zwei Antriebsrädern und einem losen Nachlaufgrad (auch Schlepprad, Towed Wheel oder Castor Wheel genannt), wobei die beiden Antriebsräder jeweils separat angetrieben werden. Ein damit ausgestattetes mobiles Fahrzeug kann einfach und beliebig manövriert werden und z.B. auf der Stelle drehen. Auch der oft eingesetzte Kettenantrieb (s. **Abb. 1.3**) entspricht einem Differentialantrieb und unterscheidet sich bei der Steuerung und Regelung der Antriebsmotoren nicht. Dennoch müssen Unterschiede beim Rad- und Kettenantrieb bezüglich Hindernis- und Fahrwegerkennung berücksichtigt werden: beispielsweise stellt das Objekt „Treppe“ für den Radantrieb ein Hindernis dar und kann für den Kettenantrieb als Fahrweg interpretiert werden.
- Der Fahr- / Lenkantrieb (s. **Abb. 1.1.a**) setzt sich ebenfalls aus drei Rädern zusammen, wobei ein Räderpaar lediglich aus nichtlenkbaren losen Rollen besteht und das dritte Rad zum Lenken sowie gleichzeitig für den Antrieb dient. Auch hier muss bei der Navigation ein Lenkradius bedacht werden.
- Als weiterer Radantrieb sei das Rocker-Bogie-Fahrwerk genannt, welches bspw. beim VolksBot-XT (eXtreme Terrain) [111] und insbesondere bei den extra-terrestrischen mobilen Robotern, wie den Mars-Rovern Pathfinder [18] [112], Spirit [19], Opportunity [19], Curiosity (s. **Abb. 1.4.b**) [20], Exomars (s. **Abb. 1.4.c**) [21] [22] [113], usw. zum Einsatz kommt. Das Rocker-Bogie-Fahrwerk besteht aus sechs Rädern, jeweils drei Rädern pro Fahrzeugseite, wobei die jeweils vorderen beiden Räder an pendelnd aufgehängten Bügeln fixiert sind. Jedes Rad wird separat angetrieben, so dass das gesamte Fahrwerk als Differentialantrieb arbeiten kann. Bei den Mars-Rovern kann darüber hinaus auch noch jedes Rad separat gelenkt werden, womit das gesamte Fahrwerk auch als Fahr- / Lenk-

antrieb realisiert werden kann. Mit Hilfe solch einer Konstruktion können kleine Hindernisse leicht überwunden und auch Treppenstufen befahren werden.

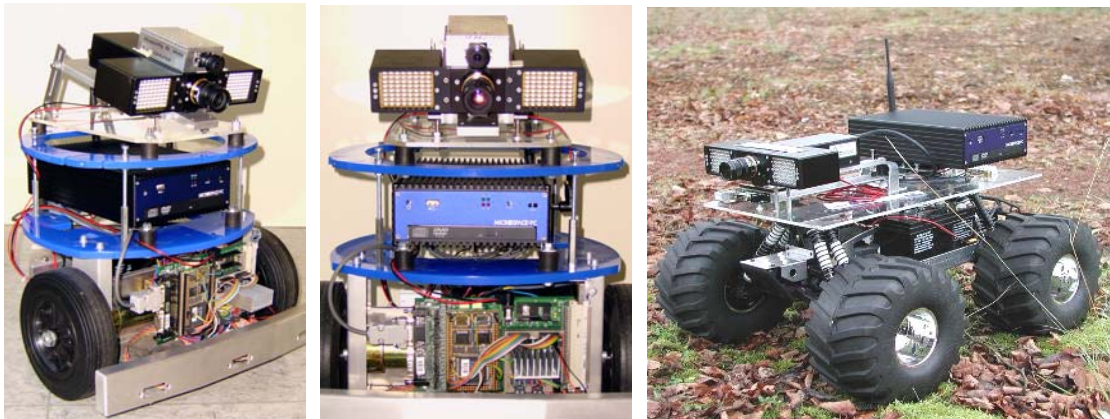


Abb. 4.1: Die mobilen Roboter Tom3D (Tele Operated Machine with 3D PMD-Camera) (l. und M.) sowie Merlin3D (Mobile Experimental Robot for Locomotion and Intelligent Navigation with 3D PMD-Camera) (r.) [82] [90] [91].

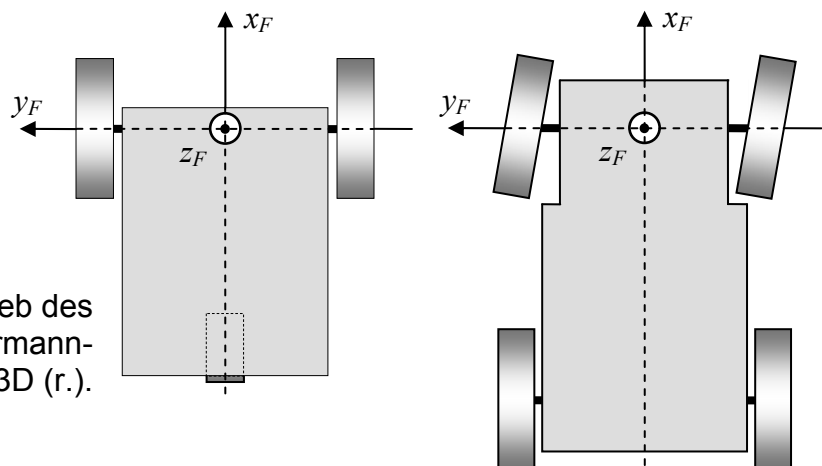


Abb. 4.2: Differentialantrieb des Tom3D (l.) und Ackermannantrieb des Merlin3D (r.).

4.2 Konventionelle und bildgebende Sensorik

4.2.1 Mechanischer Front-Bumper

Der sog. mechanische Front-Bumper stellt eine mit Sensoren bestückte Stoßstange dar, die direkte mechanische Berührungen des mobilen Fahrzeugs mit anderen Objekten bzw. Hindernissen erfassen soll. Dieser ist daher im Frontbereich, also quer zur Hauptfahrtrichtung des mobilen Fahrzeugs fixiert (s. **Abb. 4.1** l.). Die Sensoren, meist einfache Schaltkontakte (Öffner oder Schließer) können derart angeordnet werden, dass jeweils ein Sensor ein Segment der Stoßstange, z.B. linkes, mittlinkes, mittrechtes oder rechtes Segment, separat überwacht. Eine Berührung mit einem Hindernis kann so entsprechend geortet werden. Aufgrund der

nebeneinander angeordneten Segmentelemente längs der Stoßstange kann der mechanische Front-Bumper Hindernisse nur entlang einer Linie ausschließlich durch direkte mechanische Berührung, aber dort nahezu lückenlos erfassen.

Wie in Kap. 3.4 erwähnt, muss die Anzahl der Segmente stets eine gerade Zahl sein und auf vier begrenzt bleiben, um mit der Bewegungssteuerung (Kap. 4.3.5.d) und mit der Hindernisvermeidung (Kap. 5.2) zusammenarbeiten zu können. Eine Überwachung des rückwärtigen Fahrwegs ist ggf. mit einem zusätzlichen mechanischen Heck-Bumper am Heck des mobilen Fahrzeugs realisierbar, jedoch mit bsp. nur einem Segment und mit einfacher separater Hindernisvermeidung, wie z.B. Fahrzeug-Stopp bei Hindernisberührung.

4.2.2 IR-Front- und IR-Boden-Bumper

Für eine berührungslose Erfassung von Hindernissen einschließlich negativer Hindernisse werden zusätzlich noch ein IR-Front-Bumper sowie ein IR-Boden-Bumper im Frontbereich des mobilen Fahrzeugs verbaut. Der IR-Front-Bumper wurde mit IR-Sensoren vom Typ GP2D12 [110] mit 10 - 80 cm Distanzbereich und der IR-Boden-Bumper mit IR-Sensoren vom Typ GP2D120 [110] mit 4 - 30 cm Distanzbereich bestückt.

Ein IR-Sensor besteht aus einer IR-Sendediode sowie einem PSD (Position Sensitive Device)-Chip mit optischer Linse und erfasst die Hindernisdistanz anhand des reflektierten IR-Signals nach dem Triangulationsprinzip. Beide Bumper wurden derart aufgebaut, dass gemäß den oben definierten Segmenten jeweils ein IR-Sensor mit entsprechender Front- bzw. Boden-Ausrichtung platziert ist.

Mit ihren typischen keulenförmigen Sende- und Empfangscharakteristiken können IR-Sensoren nur an denjenigen Stellen Daten erfassen, wo sie positioniert und entsprechend ausgerichtet sind. Eine zu vermeidende Überlappung des Erfassungsbereichs zum Nachbarsensor ist vom Montageabstand und von der Ausrichtung abhängig und liegt, wenn überhaupt, eher im fernereren Frontbereich des mobilen Fahrzeugs. Im nahen Frontbereich hingegen bestehen Lücken im

Erfassungsbereich einer IR-Sensorkette, sodass Blindbereiche verbleiben, in denen keine Hindernisse erfasst werden können.

Generell sollen die Schaltzustände aller Sensoren, d.h. die des mechanischen Front-Bumpers, des IR-Front- und des IR-Boden-Bumpers, als digitale Signale zur Verfügung stehen. Der jeweilige Schaltersensor des mechanischen Front-Bumpers muss dafür z.B. mit einem hochohmigen Spannungsteiler beschaltet sein. Die IR-Sensoren hingegen liefern lediglich analoge Ausgangssignale, welche z.B. mit einer analogen Komparatorschaltung [114] in digitale TTL-Signale umgewandelt werden können.

Werden alle Sensorausgänge jeweils über eine Darlington-Transistor-Stufe in Open-Collector-Schaltung geführt, dann können mehrere Sensorsignale desselben Segments mit Hilfe einer Wired-OR-Verknüpfung zu einem einzelnen Signal zusammengefasst werden [114]. Damit können bsp. die drei Sensorsignale des Segments „Mittellinks“ vom mechanischen Front-Bumper, vom IR-Front-Bumper sowie vom IR-Boden-Bumper zusammengefasst und den Mikrokontroller C167 zur weiteren Datenverarbeitung übergeben werden. Für eine schnelle und sichere Fahrzeug-Stopp-Funktion, z.B. bei Personenberührung durch das mobile Fahrzeug, kann dieses Signal auch genutzt werden, um über ein Relais die Antriebsmotoren auszuschalten (s. Kap. 4.4).

4.2.3 Inklinationssensor

Der Neigungs- bzw. Inklinationssensor 3DM der Fa. Microstrain [115] ermittelt den Roll-, Pitch- und Yaw-Winkel (φ_{Roll} , φ_{Pitch} und φ_{Yaw}) in Bezug zum terrestrischen Magnetfeld und Gravitationsfeld. Damit können alle drei Rotationswinkel jeweils um die x-, y- und z-Achse des Roboterkoordinatensystems (s. Kap 6.1) des sich im unebenen Gelände bewegendes Fahrzeugs erfasst werden. φ_{Roll} misst dabei von -70° bis $+70^\circ$, φ_{Pitch} und φ_{Yaw} reichen von 0° bis 360° .

Während für die Bestimmung von Roll- und Pitch-Winkel das Erdgravitationsfeld verwendet wird, wird für die Yaw-Winkelbestimmung das Erdmagnetfeld genutzt. Da letztere Messmethode durch ferromagnetische Materialien, stromführende Leitungen usw. verfälscht werden können, soll hier der Yaw-Winkel φ_{Yaw} , der dem

Orientierungswinkel α_F des mobilen Fahrzeugs entspricht, vorzugsweise durch bildgebende Sensoren, wie die PMD-Kamera (s. Kap. 6.8), ermittelt werden.

Der Inklinationssensor 3DM kann via RS232 und per C/C++ Programm u.a. in den sog. „Command Mode“ gesetzt werden („Poll-Mode“ und „Continuous Angle Mode“ sind die weiteren Modi), um dann anschließend die drei gewünschten Winkelwerte via RS232 an den Embedded-PC zurückzusenden. Neuere Systeme ermöglichen die Kommunikation via USB-Verbindung.

4.2.4 Rad-Encoder

Beim mobilen Roboter Tom3D werden zwei Rad-Encoder für die Geschwindigkeitsregelung der beiden Antriebsmotoren verwendet und können darüber hinaus auch für die Odometrie zur Selbstlokalisierung (s. Kap. 6.5), eingesetzt werden. Ein Rad-Encoder mit $I_{Encoder_{max}} = 500$ Imp pro Umdrehung ist mechanisch direkt an die Motorwelle eines Antriebsmotors gekoppelt. Eine komplette Radumdrehung (Raddurchmesser $d_{Rad} = 160$ mm, $U_{Rad} = \pi \cdot d_{Rad}$), einschließlich Getriebeübersetzung mit $\ddot{u} = 1/38,3$, erzeugt 19.150 Rad-Encoder-Impulse und legt damit eine Wegstrecke von ca. 50,3 cm oder pro Rad-Encoder-Impuls eine Wegstrecke von ca. 26,25 μ m zurück. Die zusätzliche Nutzung der Rad-Encoder für die Odometrie bzw. für die Selbstlokalisierung ist auch sinnvoll, wenn wie in Kap. 6 dargestellt, mehrere Selbstlokalisierungsmethoden zur Verfügung stehen, deren Daten mit Hilfe von Kalman Filter (KF) bzw. Extended Kalman Filter (EKF) fusioniert werden, um genauere Werte zu bekommen [116] - [121].

4.2.5 2D-Kamera

Eine 2D-Kamera, mit einem CCD- oder einem CMOS-Chip als bildgebendem Sensor ausgestattet, detektiert pixelweise Lichtintensitäten vieler optischer Raumprojektionen. Die Beleuchtung bei solchen Kameras ist meist passiv, im Qualitätssicherungsbereich ist sie aktiv und im Sicherheitsbereich ist sie zusätzlich gegen Fremdlicht abgeschirmt. 2D-Kameras können ohne Objektiv als sog. Lochkameras, mit Normalobjektiv oder mit Fischaugenobjektiv arbeiten.

In der mobilen Robotik dient die 2D-Kamera zur visuellen Kontrolle für den Operator bei Teleoperationen [77]. Dies schließt auch teilautonome Andockmanöver eines mobilen Fahrzeugs und teilautonomes Greifen von Gegenständen mittels Manipulator ein. Des Weiteren wird sie zum Entdecken und Vermessen von künstlichen Land- oder Referenzmarken verwendet (s. Kap. 6.3) [122] [123].

In dieser Arbeit ist eine 2D-Kamera zusammen mit einer PMD-Kamera, parallel zu deren optischen Achse und in Fahrtrichtung angeordnet, in das mobile Fahrzeug integriert (s. Kap. 3.1 und **Abb. 4.1**). Mit beiden Kameras soll insbesondere der Frontbereich des mobilen Fahrzeugs erfasst werden. Hiermit können die Videobilder beider Kameras einerseits separat als 3D- und als 2D-Videobild oder andererseits zusammen als 3D/2D-Videobild-Kombination verwendet werden (s. Kap. 2.5) [23] [36] [46] [52] [75] [77] - [84]. Entsprechend Kap. 6.6 kann die 2D-Kamera auch für die visuelle 2D-Odometrie zum Einsatz kommen.

4.2.6 Omnidirektionale Kamera

Eine omnidirektionale Kamera ermöglicht die Erfassung der gesamten Umgebung des mobilen Fahrzeugs als 360°-Panoramabild [137]. Panoramaaufnahmen mit Hilfe einer einfachen 2D-Kamera durch Fahrzeugbewegungen oder durch eine Drehvorrichtung und Zusammensetzen der Teilbilder entfallen hierbei.

Eine solche Kamera ist in folgender Weise aufgebaut: Über eine nach oben senkrecht aufgerichtete Kamera wird ein verspiegelter Kegel fixiert. Die Abbildung der Umgebung innerhalb der Kamera erfolgt stets über die verspiegelte Oberfläche des Kegels (**Abb. 4.3**). Das mit dieser Kamera gewonnene Bild muss entsprechend der Kegelkontur entzerrt werden.

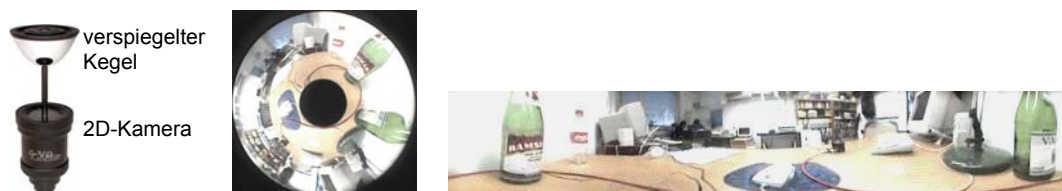


Abb. 4.3: Omnidirektionale Kamera (l.), deren omnidirektionale (M.) und entzerrte Aufnahme (r.).

4.2.7 Stereovision

Im Allgemeinen haben 2D-Kameras den Nachteil, dass durch die Projektion des Umgebungsbildes auf den Bildsensor die Tiefeninformationen verloren gehen. Durch Passiv-, Aktiv- oder Motion-Stereo-Verfahren lassen sich diese Tiefeninformationen allerdings nur mit hohem rechnerischem Aufwand teilweise rekonstruieren. Stereovision mit Hilfe zweier 2D-Kameras bzw. Stereo-Doppelvideobilder führt eine indirekte Abstandsmessung über Winkelpeilung und Triangulation durch:

- Ein Passiv-Stereovision-Kamerasystem besteht aus zwei im definierten Abstand nebeneinander und parallel zu ihren optischen Achsen fixierten 2D-Kameras, aus deren Stereo-Doppelvideobildern die Tiefeninformation für jedes Pixel errechnet wird. Dabei wird innerhalb des Stereo-Doppelvideobildes nach korrespondierenden 2D-Bildelementen gesucht, z.B. auf eine hinreichend grobe Textur, um über bspw. das Verfahren der 2D-Korrelation den Abstand zu berechnen [124].
- Ein Aktiv-Stereovision-Kamerasystem besteht aus zwei im definierten Abstand fixierten und jeweils separat schwenkbaren 2D-Kameras. Ähnlich dem menschlichen Sehen erfassen beide 2D-Kameras die Szene und suchen unter synchroner mechanischer Fokussierung und Fixierung (Akkomodation und Adaption) geeignete Pixelcluster. Daraus generieren sie quasi fortlaufend Triangulationsdreiecke, aus denen die Tiefendaten berechnet werden [124].
- Bei Motion-Stereovision wird lediglich eine einzelne 2D-Kamera auf das mobile Fahrzeug in dessen Fahrtrichtung fixiert. Während der Fahrt werden stets zwei aufeinander folgende Videobilder mit einem Stereo-Verfahren ausgewertet. Im Gegensatz zum o.g. Passiv- und Aktiv-Stereovision-Verfahren sind hier die Kamerapositionen und die Distanz zwischen zwei aufeinander folgenden Videobildaufnahmen von der Fahrtrichtung sowie -geschwindigkeit abhängig [124].

4.3 Hard- und Software-Architektur

Der in **Abb. 4.1** gezeigte mobile Roboter Tom3D wurde auf der Basis des MOSRO aufgebaut, der ursprünglich von der Fa. Robowatch [8] als Bewachungsroboter entsprechend **Abb. 1.1.f** konzipiert wurde. Als Fahrwerk verfügt Tom3D über den

o.g. Differentialantrieb (**Abb. 4.2** l.), bestehend aus zwei separaten Antriebssystemen mit je einem Rad (\varnothing 160 mm), dem Gleichstromgetriebemotor GM9236 der Fa. Pittman [125] mit 12 V, ca. 50 W, Getriebeübersetzung $\ddot{u} = 1/38,3$ und Rad-Encoder (500 ppr), sowie einem losen Nachlaufgrad (\varnothing 75 mm).

Die beiden Rad-Encoder dienen zur Geschwindigkeitsregelung (s. Kap. 4.3.2) und zur groben Posebestimmung (s. Kap. 6.5). Für die Hindernis- und Fahrwegerkennung wird die PMD-Kamera mit einer Auflösung von 64(h) x 48(v) Pixeln [34] eingesetzt. Des Weiteren ist Tom3D mit einem Mikrocontroller C167 [126] [127] für die Geschwindigkeitsregelung, Bewegungssteuerung (s. Kap. 4.3.5.d) und Hindernisvermeidung (s. Kap. 5.2), sowie mit einem Embedded-PC für die Bildverarbeitung und die Kommunikation ausgestattet (**Abb. 4.4**). Die Kommunikation bzw. der Zugriff des Operators auf das mobile Fahrzeug ist via WLAN, ggf. auch via Mobil- oder Satellitentelefonmodem möglich (s. Kap. 4.3.5.a).

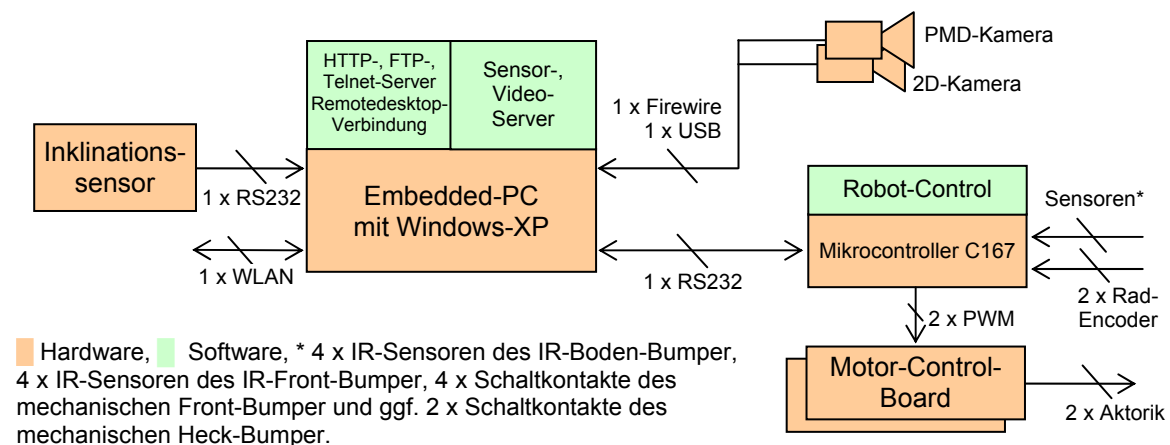


Abb. 4.4: Hard- und Software-Komponenten von Tom3D und Merlin3D.

Tom3D soll ausschließlich innerhalb eines Gebäudes mit ebener Flächenausdehnung (Indoor-Areal) agieren. Für den Außenbereich (Outdoor-Gelände) wurde der mobile Roboter Merlin3D (s. **Abb. 4.1** r.) mit Ackermannantrieb ausgestattet. Er ist, bis auf die Motor-Controller-Einheit, mit gleicher Systemarchitektur wie Tom3D aufgebaut. Um Neigungen und Steigungen des im unebenen Gelände fahrenden mobilen Fahrzeugs zu erfassen, wurde hier zusätzlich ein Inklinationssensor [115] eingesetzt. Da GPS bzw. DGPS nur außerhalb von Gebäuden einsatzfähig ist, können diese Sensoren auch nur im Outdoor-Bereich zur absoluten Selbstlokalisierung verwendet werden.

4.3.1 Motor-Control-Board

Als Hardware für die Ansteuerung der Antriebsmotoren der mobilen Roboter Tom3D und Merlin3D wurden die Motor-Control-Boards MD03 und MD22 der Fa. Devantech [128] [129] gewählt. MD22 besitzt zwei H-Brücken [114] und kann somit die beiden Gleichstromgetriebemotoren GM9236 [125] des Tom3D separat betreiben. Dagegen verfügt MD03 nur über eine H-Brücke und kann damit den Antriebsmotor des Merlin3D ansteuern. Während die Lenkung beim Tom3D mit Differentialantrieb aus den unterschiedlichen Drehzahlen der Antriebsmotoren resultiert, wird stattdessen beim Merlin3D mit Ackermannantrieb ein Servo-Lenkantrieb ohne zusätzliches Motor-Control-Board eingesetzt.

Bei allen Motor-Control-Boards kann die Drehzahl der Antriebsmotoren bzw. der Stellwinkel des Servo-Lenkantriebs u.a. mittels PWM-Signal mit Mittelstellung variiert werden. Mittelstellung mit z.B. $PWM_{50\%}$ bedeutet dabei, dass bei einem Tastverhältnis des PWM-Signals von 50% die Antriebsmotoren still stehen bzw. der Servo-Lenkantrieb in Null-Stellung verharrt, bei $PWM < 50\%$ rückwärts dreht bzw. links lenkt und bei $PWM > 50\%$ vorwärts dreht bzw. rechts lenkt. Alle PWM-Eingangssignale für die Motor-Control-Boards und den Servo-Lenkantrieb werden vom Mikrocontroller C167 erzeugt und im TTL-Standard übermittelt.

4.3.2 Geschwindigkeitsregelung

Der mobile Roboter Tom3D soll sich beim Navigieren sowohl durch ein ebenes als auch durch ein geringfügig unebenes 2D-Areal stets mit konstanter Geschwindigkeit bewegen können. Dies verlangt, dass seine beiden Gleichstromgetriebemotoren GM9236 [125] auch bei leichter Steigung oder leichtem Gefälle immer die von der Hindernisvermeidung (Kap. 5.2) vorgegebene Geschwindigkeit konstant beibehalten. Daher wurde eine Geschwindigkeitsregelung separat für jeden Antriebsmotor auf der Basis der entsprechenden Rad-Encoder-Daten realisiert.

In **Abb. 4.5.a** ist die Sprungantwort bei einem Eingangssprungsignal von $U_{in} = 6\text{ V}$ eines der beiden baugleichen Antriebsmotoren dargestellt. Das Getriebeispiel (Getriebe lose) stellt eine Totzeit von $T_t \approx 0,066\text{ s}$ dar. Im Übrigen ist ein PT1-Verhalten

mit $T_S \approx 0,243$ s und einem Maximalwert der Sprungantwort von $I_{Encoder} = 808 \text{ Imp}/T_{abstast}$, d.h. die Impulszahl des Rad-Encoders auf der Motorwelle, festzustellen. Bei einer Abtastzeit des Reglers von $T_{abstast} = 26,214$ ms ergibt dies eine Streckenverstärkung von $K_S = I_{Encoder}/U_{in} = 808 \text{ Imp}/6 \text{ V} = 134,67 \text{ Imp}/\text{V}$.

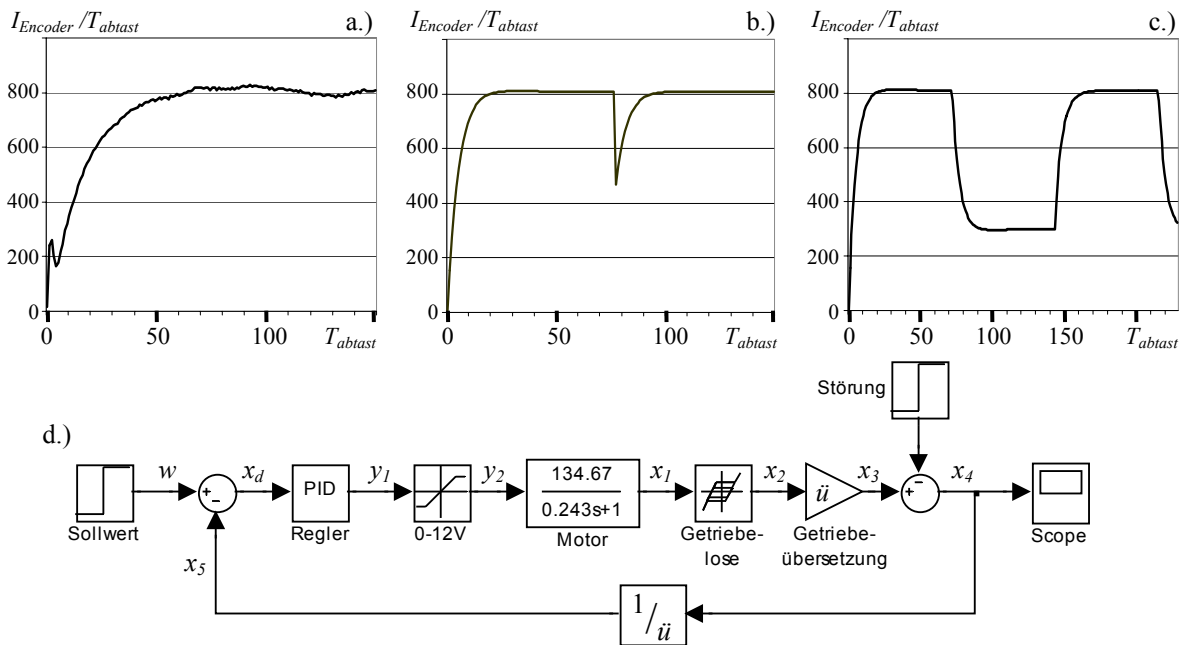


Abb. 4.5: a.) PT1-Verhalten eines Antriebsmotors, b.) die Sprungantwort und c.) Antwort einer Sollwertfolge mit Hilfe d.) einer PID-Geschwindigkeitsregelung.

Für eine Geschwindigkeitsregelung kann mit Hilfe der Optimierungsregeln von Ziegler und Nichols [131], sowie unter Berücksichtigung der Abtastzeit nach Takahashi [131] für solch eine PT1-Regelstrecke sowohl ein PI-Regler als auch ein PID-Regler eingesetzt werden [130] - [132]. Für Letzteren lassen sich folgende Parameter errechnen (4.1) (4.2):

$$K_R = \frac{1,2 \cdot T_S}{K_S \cdot (T_t + T_{abstast})}, \quad T_N = \frac{2 \cdot \left(T_t + \frac{T_{abstast}}{2}\right)^2}{T_t + T_{abstast}} \quad \text{und} \quad T_V = 0,5 \cdot (T_t + T_{abstast}) \quad (4.1)$$

$$K_P = K_R = 0,0236 \text{ V}/\text{Imp}, \quad (4.2a)$$

$$K_I = \frac{K_P}{T_N} = \frac{0,0236 \text{ V}}{0,1348 \text{ Imp} \cdot \text{s}} = 0,1747 \text{ V}/\text{Imp} \cdot \text{s} \quad \text{und} \quad (4.2b)$$

$$K_D = K_P \cdot T_V = 0,0236 \cdot 0,0459 \text{ V} \cdot \text{s}/\text{Imp} = 0,0011 \text{ V} \cdot \text{s}/\text{Imp} \quad (4.2c)$$

Der Regelkreis zur Drehzahlregelung eines einzelnen Antriebsmotors wurde hier zunächst mit Hilfe der Software SIMULINK [133] und mit den in (4.2) errechneten Parametern simuliert (s. **Abb. 4.5.d**). Aufgrund der Spannungsbegrenzung von 0 - 12 V innerhalb des Regelkreises entstehen Überschwingungen in der Sprungantwort,

sodass insbesondere der Integrationsparameter auf $K_I = 0,05 \text{ V/Imp}\cdot\text{s}$ nachjustiert werden musste. Anschließend wurde der PID-Regler als Software-Algorithmus innerhalb des Mikrocontrollers C167 implementiert.

In **Abb. 4.5.b** ist das Befahren einer Steigung in Form einer Störung zu sehen, die durch den PID-Regler schnell ausgeglichen wird. **Abb. 4.5.c** hingegen zeigt das Geschwindigkeits- bzw. Drehzahlverhalten des Antriebsmotors aufgrund wechselnder von der Hindernisvermeidung vorgegebener Sollwerte an. Mit Hilfe des PID-Reglers werden die Istwerte schnell diesen Vorgaben angepasst.

4.3.3 Sensor-Server

Das Software-Programm „Sensor-Server“ arbeitet auf dem Embedded-PC unter Windows-XP. Im Gegensatz zu einem echtzeitfähigen Betriebssystem, wie z.B. VxWorks [134], gilt Windows-XP als nicht-echtzeitfähiges Betriebssystem. Dennoch kann auch unter WinXP mit Hilfe der Anweisung „start /realtime /B programm.exe“, in der u.a. die Option „realtime“ oder „high“ enthalten sein muss, die Anwendung in der REALTIME- bzw. HIGH-Prioritätsklasse gestartet werden. Windows XP kann aber auch durch eine Echtzeiterweiterung, wie z.B. CeWin [250] oder Kithara RealTime Suite [251], ergänzt werden, so dass o.g. Programm in einer Echtzeitumgebung eingebettet werden kann. Grundsätzlich hat Windows-XP den Vorteil, dass hierfür sämtliche Software-Treiber für die im mobilen Fahrzeug verbauten Sensoren, wie Inklinationssensor, 2D-Kameras (2D-Kamera, Omnidirektionalkamera, Stereovision), PMD-Kamera, ggf. GPS bzw. Galileo, usw. verfügbar sind⁶.

Das Software-Programm „Sensor-Server“ auf dem Embedded-PC unter Windows-XP arbeitet in einer Schleifenstruktur mit folgenden Arbeitsschritten:

- Daten der Sensoren, wie PMD-Kamera, Inklinationssensor, usw. einlesen
- Sensordaten ggf. aufbereiten und filtern (s. Kap. 2.4)
- Hindernis- und Fahrwegerkennung durchführen (s. Kap. 3.4)
- zwecks Hindernisvermeidung (s. Kap. 5.2) Hindernisdaten via RS232 an den Mikrocontroller C167 senden.

⁶ Bezüglich eines Echtzeit-Betriebssystems auf der Basis von Linux sei auf [252] verwiesen.

Unter Berücksichtigung der Integrationszeit der PMD-Kamera (s. Kap. 2.3.1), der Median-Filterung (s. Kap. 2.4) und der auf jedes PMD-Videobild angewandten o.g. Arbeitsschritte reduziert sich die Frame-Rate der PMD-Kamera auf ca. 10 Bilder pro Sekunde.

4.3.4 Video-Server

Eine besondere Bedeutung für die Telepräsenz kommt der Visualisierung durch Kameras zu. Eine oder sogar mehrere Kameras sollen dem Operator eine Sicht der Fahrzeugumgebung via Internet geben [77]. U.a. die in Kap. 4.2.5 bis 4.2.7 beschriebenen Kameras kommen hierfür in Frage. Vorzugsweise sollten Kameras mit USB (Universal Serial Bus)-Schnittstelle, Version 2.0, mit einer Datenrate von bis zu 480 MBit/s gewählt werden, da solche Kameras im Anschaffungspreis günstig sind und viele Bildverarbeitungsbibliotheken, wie z.B. OpenCV [45], ICE [135], usw. dafür entsprechende Datenzugriffsfunktionen bieten.

Neben den Datenzugriffsfunktionen enthalten solche Bibliotheken auch Bildverarbeitungsfunktionen, die insbesondere dann von Bedeutung sind, wenn die Kameras über die Umgebungsvisualisierung hinaus auch zur Suche von künstlichen Landmarken bzw. Referenzmarken eingesetzt werden sollen (s. Kap. 6.3).

Im einfachsten Fall wird eine USB-Web-Kamera in Fahrtrichtung des mobilen Fahrzeugs, ggf. mit einem intuitiv gewählten Neigungswinkel, am Chassis fixiert, um dessen nahen und mittleren Frontbereich aufzunehmen. Zur Vergrößerung des Sichtbereichs eignet sich eine sog. Fischaugenkamera [33] mit Weitwinkelobjektiv oder eine schwenkbare 2D-Kamera, z.B. die schwenkbare USB-Web-Kamera der Fa. Axis [136]. Eine omnidirektionale Kamera [137] könnte mit Blickrichtung auf das Fahrzeug am höchsten Punkt des Chassis befestigt werden und kann dann alle Bereiche der Fahrzeugumgebung einsehen, einschließlich einer Draufsicht auf das mobile Fahrzeug selbst.

Ein brauchbares Visionsystem zwecks Telepräsenz und Kontrolle eines mobilen Fahrzeugs via Internet stellt eine Kombination aus einer USB-Web-Kamera und zusätzlich einer omnidirektionalen Kamera dar. Mit Letzterem steht dem Operator eine

mäßig aufgelöste Rundumsicht der gesamten Fahrzeugumgebung einschließlich einer Fahrzeugdraufsicht zur Verfügung. Darüber hinaus bekommt er über die USB-Web-Kamera ein hinreichend hoch aufgelöstes Videobild des nahen und mittleren Frontbereichs, also des Fahrweg- und Aktionsbereichs des mobilen Fahrzeugs.

Zur Bereitstellung der Videodaten via Internet muss für jede eingesetzte Kamera, wie USB-Web-Kamera, Fischaugenkamera, omnidirektionale Kamera, usw. ein separater Video-Server bereitgestellt werden, der ebenfalls auf dem Embedded-PC unter Windows-XP läuft. Der Video-Server hat die Aufgabe, bei Anfrage durch den Operator, Videodaten der Kamera über die USB-Schnittstelle auf den Embedded-PC zu laden und schließlich via WLAN, UMTS oder Satellitenmodem kontinuierlich an die GUI des Operators zu senden.

Als Software des Video-Servers reicht im einfachsten Fall ein schlichter WebCam-Server aus, der allerdings nur im Sekundentakt Videobilder der USB-Web-Kamera an den Operator senden kann. Sofern die Übertragungsraten es erlauben, ist dafür auch ein Video-Server mit Streaming-Technik, der Videodaten als kontinuierlichen Datenstrom übermitteln kann, besser geeignet (s. Kap. 4.3.5.e, **Abb. 4.11**). Im Falle einer Fischaugenkamera oder einer omnidirektionalen Kamera muss der betreffende Video-Server zusätzlich noch jedes Videobild entsprechend der Fischaugenoptik bzw. der Spiegelkontur entzerren.

4.3.5 Vernetzung, Telemetrie und Kommunikation

4.3.5.a Topologie eines vernetzten mobilen Fahrzeugs

Die Datenkommunikation innerhalb des mobilen Fahrzeugs vollzieht sich zwischen Mikrocontroller C167 und Embedded-PC via RS232. Außerhalb des mobilen Fahrzeugs erfolgt sie zwischen Embedded-PC und Operator via Wireless-LAN, Mobil- oder Satellitenfunk (**Abb. 4.6 - 4.10**). Für die Kontrolle des mobilen Fahrzeugs durch den Operator ist also die sog. „äußere“ Datenkommunikation via Wireless-LAN, Mobil- oder Satellitenfunk von Bedeutung. Über diese soll der Operator einerseits Videobilder der Fahrzeugumgebung erhalten und andererseits soll zur Kontrolle des mobilen Fahrzeugs eine Benutzerschnittstelle (GUI) auf seinen Laptop

geladen werden können (**Abb. 4.11** bis **4.13**). Für Letztere muss auf dem Embedded-PC neben dem o.g. Video-Server (s. Kap. 4.3.4) auch ein WWW- bzw. Http (Hypertext Transfer Protocol)-Server laufen, der die GUI in Form von Web-Pages via Internet zur Verfügung stellen kann (s. Kap. 4.3.5.e).

Das Wireless-LAN mit dem Standard IEEE 802.11b/g [138] [139] überträgt Daten mit einer Datenrate bis 54 MByte/s bei einer Reichweite bis ca. 140 m. Auf dieser Distanz lässt sich zwischen Operator und mobilem Fahrzeug mit einfachen Mitteln ein sog. Ad-hoc-Wireless-LAN (**Abb. 4.6**), auch IBSS (Independent Basic Service Set) -Topologie [139] genannt, einrichten. Hierzu müssen lediglich die im Embedded-PC des mobilen Fahrzeugs und die auf dem Desktop-PC, Laptop, PDA (Personal Digital Assistant) oder Smartphone des Operators vorhandene WLAN-Module entsprechend konfiguriert werden.

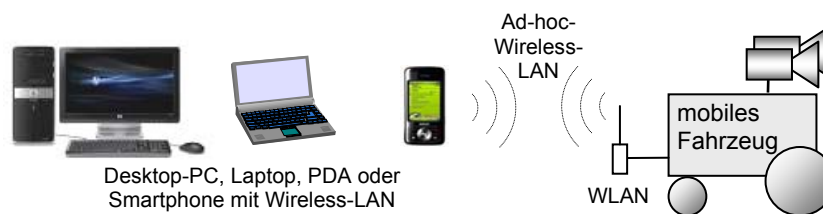


Abb. 4.6: Vernetzung über Ad-hoc-Wireless-LAN.

Eine Erweiterung zum Ad-hoc-WLAN-Konzept stellt die in **Abb. 4.7** gezeigte Variante eines sog. BSS (Basic Service Set) - Netzwerks [139] dar. In einer bereits vorhandenen LAN-Infrastruktur kann über einen zusätzlichen WLAN-Access-Point eine Verbindung zum mobilen Fahrzeug aufgebaut werden. Der Vorteil hierbei liegt darin, dass z.B. Planungsaufgaben, wie Pfad- und Aufgabenplanung, auf Rechnern des LAN durchgeführt werden können, ohne dass das mobile Fahrzeug in Betrieb sein muss (Offline-Betrieb).

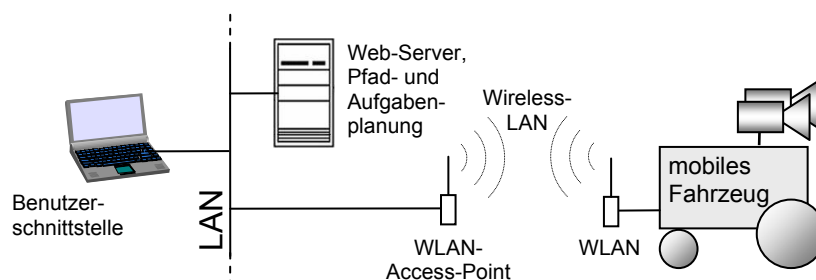


Abb. 4.7: Vernetzung über LAN, Access-Point und Wireless-LAN (Infrastruktur-WLAN).

Eine Steigerung der Datenübertragungsentfernung kann mit dem Standard IEEE 802.11h [139] mit höherer Sendeleistung bis ca. 1000 mW und durch den Einsatz externer Stabantennen oder gar Richtfunkantennen auf > 2 km in freier un bebauter Umgebung erreicht werden. Durch den Einsatz eines oder mehrerer WLAN-Access-Points als WLAN-Repeater in der Form eines sog. WDS (Wireless Distribution System) [139] ist ebenfalls eine Distanz von > 2 km überbrückbar. Mit dem sog. WiMAX (Standard IEEE 802.16 und insbesondere IEEE 802.16e) kann für mobile Anwendungen [139] [140] eine Reichweite bis ca. 5 km bei freier Sicht und eine Datenübertragungsrate bis ca. 15 Mbit/s erlangt werden.

Eine weitere Möglichkeit zur Erhöhung der Distanz zwischen Operator und mobilem Fahrzeug ist mit Hilfe des Telefonfestnetzes auf der Basis von DSL (Digital Subscriber Line) [139] in Verbindung mit einem WLAN-Access-Point erreichbar. Entsprechend **Abb. 4.8** bekommt der Operator via Internet, Gateway, DSL-Telefonfestnetz und Wireless-LAN einen Zugang auf den Embedded-PC des mobilen Fahrzeugs. Auch hier kann wahlweise ein Ad-hoc-WLAN oder ein Infrastruktur-WLAN eingesetzt werden. Diese Variante ist für Teleroboteranwendungen insbesondere in privaten Haushalten mit DSL-Anschluss und für den Datenzugriff außer Haus von Bedeutung.

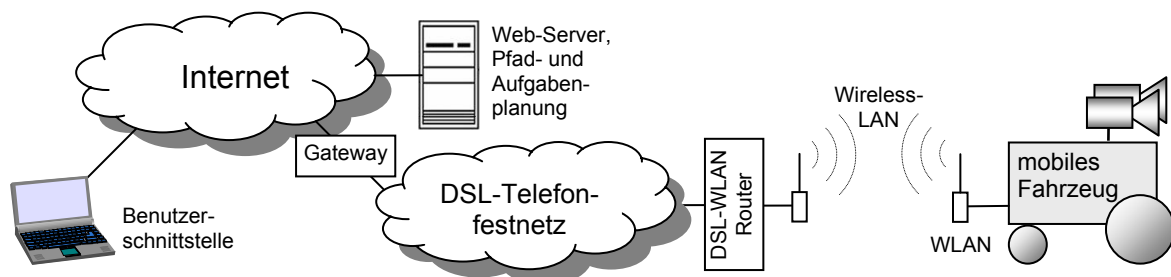


Abb. 4.8: Vernetzung über DSL und Wireless-LAN.

In Regionen ohne DSL oder gar in freier Landschaft geben terrestrische Mobilfunknetze weitere Möglichkeiten der Teleoperation. Im Handel sind diverse Mobilfunkmodem-Module erhältlich, die in den Embedded-PC des mobilen Fahrzeugs integriert werden können. Auch bieten verschiedene Mobilfunkbetreiber hierfür sog. USB-Internet-Sticks [141], auch Web- oder Surf-Stick genannt, die einen Embedded-PC an ein Mobilfunknetz anbinden können, um dann über einen Gateway ins Internet zu gelangen (**Abb. 4.9**). Für eine zeitweise bzw. permanente Verbindung muss ein solches Mobilfunkmodem als sog. Prepaid- bzw. Flatrate-Verbindung eingerichtet

sein. Je nach Netzabdeckung erfolgt die Datenkommunikation über die Mobilfunkstandards GPRS (53,6 kbit/s Download und 26,8 kbit/s Upload), EDGE (220 kbit/s und 110 kbit/s), UMTS (384 kbit/s), UMTS Broadband (HSDPA 7,2 Mbit/s und HSUPA 3,5 Mbit/s) oder LTE (35 Mbit/s und 8 Mbit/s) [142].

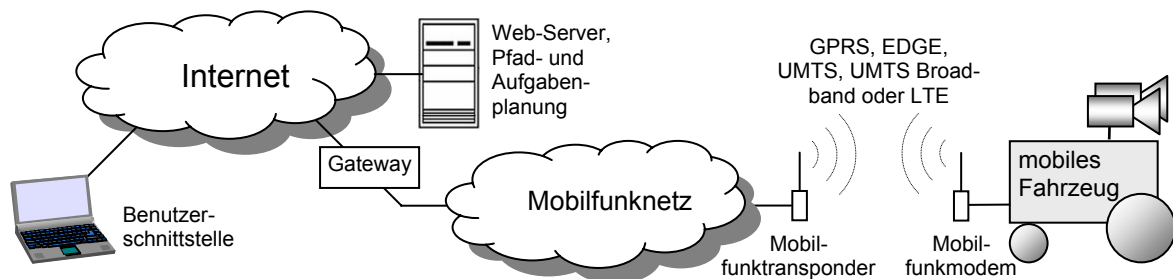


Abb. 4.9: Vernetzung über Mobilfunknetze.

Eine Datenübertragung beispielsweise via EDGE von Videobildern im GIF-Format [143] einer am mobilen Fahrzeug fixierten WebCam mit minimal 160 x 120 Pixel und nur einem Bild pro Sekunde erfordert eine Datenübertragungsrate von $256 \times 160 \times 120 \text{ Bit} / 1\text{s} \approx 4,7 \text{ Mbit/s}$. Mit Hilfe geeigneter Komprimierungsverfahren, z.B. LZW (Lempel-Ziv-Welch-Algorithmus) [144], müsste die Datenmenge um ca. 95% reduziert werden, um auf z.B. 220 kbit/s zu kommen. Mit der Nutzung der Mobilfunkstandards UMTS oder UMTS Broadband stehen weitaus höhere Datenübertragungsraten zur Verfügung bei geringerem Komprimierungsaufwand; allerdings mit weniger Netzabdeckung, insbesondere fernab von größeren Städten und Metropolen.

Bei fehlender Mobilfunk-Netzabdeckung eignet sich für die Datenübertragung die Satellitenkommunikation. Hier wird eine Datenverbindung zwischen einem Satellitentelefonmodem und einem Satelliten aufgebaut. Dann folgt eine Weiterverbindung entweder direkt zu einer Bodenstation oder zu weiteren Satelliten mittels Intersatellitenlink (ISL) [145]. Von einer Bodenstation gelangt man schließlich über einen Gateway in ein Telefonnetz bzw. Internet (**Abb. 4.10**). Satellitentelefonbetreiber sind u.a. Iridium [146], Inmarsat [147], Globalstar [148], Thuraya [149], Ellipso [150], ACeS [151], Orbcomm [152], usw.

Die stationären Parabolantennen von Satellitentelefonmodems sind untauglich für mobile Fahrzeuge, da sie aufgrund der Fahrbewegung manuell stets neu auf den Satelliten ausgerichtet werden müssten. Abhilfe schafft hier z.B. das

Satellitentelefonmodem „Explorer 325“ der Fa. Thrane & Thrane [153], bei dem die Parabolantenne der Fahrzeugbewegung entsprechend automatisch nachgeführt wird. Satellitentelefonmodems mit Stabantennen, wie z.B. ein Iridium Satellitentelefon [154], sind als Satellitentelefonmodem nur bedingt tauglich, da sie mit ihrer 360°-Abstrahlcharakteristik nur wenig Sendeleistung an den Satelliten abgeben können und somit nur niedrige Datenübertragungsraten bieten.

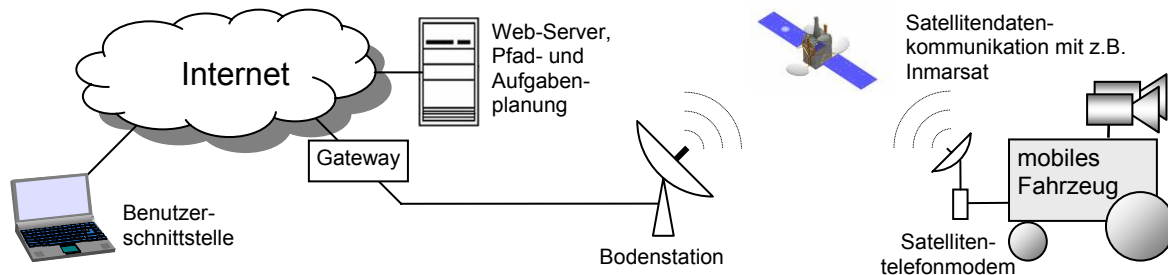


Abb. 4.10: Vernetzung über Satellitentelefonmodem.

4.3.5.b Teleaktivierung und -deaktivierung des mobilen Fahrzeugs

Im Normalfall arbeiten mobile Fahrzeuge stets im Akku-Betrieb. Dies erfordert einen sparsamen Umgang mit der in den Akkumulatoren mitgeführten elektrischen Energie. Im Ruhezustand sollten deshalb alle ungenutzten Verbraucher des mobilen Fahrzeugs ausgeschaltet sein und für dessen Inbetriebnahme entweder manuell oder per Teleaktivierung wieder eingeschaltet werden können. Vor allem die Teleaktivierung eines mobilen Fahrzeugs via Internet verlangt spezielle Maßnahmen des Datenverbindungsaufbaus, insbesondere, wenn der Embedded-PC selbst deaktiviert ist.

Ein sparsamer Umgang mit der elektrischen Energie der Akkumulatoren kann zunächst dadurch erreicht werden, dass der Embedded-PC mit einem energiesparenden Prozessor ausgestattet ist. Derzeit steht hierfür beispielsweise der Prozessor „Atom Z530“ der Fa. Intel [155] zur Verfügung. Bei einer Taktrate von 1,6 GHz entwickelt dieser Prozessor eine thermische Verlustleistung, auch TDP (Thermal Design Power) genannt, von lediglich 2,2 W [156]. Solch ein Prozessor, z.B. in ein Mini-, Nano-, Pico-, PC104-, 3,5"- oder 2,5"-Board verbaut und mit dem „US15W“-Chipsatz [156], sowie allen nötigen PC-typischen Schnittstellen versehen, benötigt insgesamt nur ca. 5 - 7,5 W im Normalbetrieb und ca. 0,1 W im Suspend-Betrieb. Damit eignet sich der „Atom Z530“ für mobile Systeme im Akkubetrieb.

Alle Systeme des mobilen Fahrzeugs, wie Motoren, Motor-Controller-Board, Mikrocontroller C167, usw. lassen sich z.B. über die Digital-I/O-Ports des Embedded-PCs und mit Hilfe von Relais per Software ein- bzw. ausgeschaltet. Dies kann der Operator jederzeit z.B. über eine Remotedesktop-Verbindung zum Embedded-PC vornehmen. Auch der Embedded-PC selbst kann über das konfigurierbare Energie-Management, dem sog. ACPI (Advanced Configuration and Power Interface) [157] des WinXP-Betriebssystems, in verschiedene Betriebsmodi versetzt werden.

Die ACPI-Spezifikation sieht dafür die Zustände G0 für Arbeits-, G1 für Schlafzustand, G2 für „Soft-Off“-Zustand und G3 für „Mechanical off“-Zustand vor [158]. Während des G1-Zustands kann das System mit S1 bis S4 (s. **Tab. 4.1**) unterschiedlich „tief schlafen“. Einzelne Geräte im System können die Zustände D0 (angeschaltet) bis D3 (ausgeschaltet) einnehmen. Und auch der Prozessor kann sich innerhalb verschiedener Zustände C0 bis C3 befinden.

S0	Working- oder Away-Mode	Normaler Betrieb
S1	Sleep-Mode	Monitor wird ausgeschaltet, CPU kann C1- bis C3-Modus einnehmen.
S2	Extended Sleep-Mode	Weitere Komponenten werden ausgeschaltet.
S3	Standby-Mode, Suspend-to-RAM (STR) o. Suspend-to-Memory (STM)	Betriebszustand des Systems wird auf das RAM gespeichert.
S4	Hibernation-Mode oder Suspend-to-Disk (STD)	Betriebszustand des Systems wird auf die Festplatte gespeichert.
S5	Soft-Off-Mode	Das System ist ausgeschaltet, nur die Spannungsversorgung durch das Netzteil bleibt bestehen, um mit der Einschalttaste einen Neustart zu ermöglichen. Andere Komponenten können ebenfalls eingeschaltet bleiben, um z.B. mit einer Netzwerkschnittstelle und mit „Wake On LAN“ den Rechner aufzuwecken.

Tab. 4.1: Betriebsmodi S0 - S5 des ACPI-Standards [157] [158].

Die in **Tab. 4.1** enthaltenen Betriebsmodi S0 bis S5 können durch den Operator auch mit Hilfe der betriebssystemeigenen WinXP-Programmen „rundll32.exe“ und „shutdown.exe“ via Remotedesktop-Verbindung und Internet initiiert werden. Dafür muss der Embedded-PC sich aber im Betriebsmodus S0 befinden. Denn ein sich im Ruhezustand (S2-, S3- und S4-Modus, s. **Tab. 4.1**) befindlicher oder gar ausgeschalteter Embedded-PC (S5-Modus, s. **Tab. 4.1**) kann nicht mehr via Kommunikationsmodul kontaktiert werden. Während die Fern- bzw. Teledeaktivierung via Internet mit o.g. Mitteln möglich ist, bedarf es für die Fern- bzw.

Teleaktivierung via Internet zusätzlicher Ausstattungen und Funktionen innerhalb der Hard- und Software des Embedded-PCs.

Die bereits für LAN bekannte Funktion „Wake on LAN“ [159], also das Aufwecken von Rechnern innerhalb eines drahtgebundenen LAN, steht mit „Wake on Wireless“ oder „Wake on Theft“ auch für WLAN zur Verfügung [160] [161] [162]. Dazu bietet z.B. der Chiphersteller Atheros u.a. den WLAN-Chip „AR9280“ an, der die „Wake on Wireless“-Funktion integriert hat [163]. Dieser WLAN-Chip findet sich auf PCIe (Peripheral Component Interconnect Express)-Modulen für WLAN diverser Hersteller. Auch für Mobilfunk-Datenverbindungen ist bspw. bei der Fa. Ericsson das PCIe-Modul „F5521gw“ mit „Wake on Wireless“-Funktion erhältlich [164].

Diese PCIe-Module sind auch im ausgeschalteten Zustand (S5-Modus) des Embedded-PC weiterhin mit der Spannungsversorgung verbunden, und zwar über den permanent aktiven ATX (Advanced Technology Extended)-Spannungsversorgungsstrang, der stets über die PCI- bzw. PCIe-Schnittstelle zur Verfügung steht. Zur Nutzung der „Wake on Wireless“-Funktion muss innerhalb des Embedded-PC die PCIe-Schnittstelle selbst mit der ACPI-Technik [157] [158] ausgerüstet sein. Ein vom Operator via WLAN bzw. Mobilfunk gesendetes Aufweck-Datenpaket, auch „Magic-Packet“ genannt, kann jetzt von solch einem PCIe-Modul interpretiert werden und anschließend den Embedded-PC vom S5-, S4-, S3-, S2- oder S1-Modus in den Aktivmodus S0 versetzen. Somit können mobile Fahrzeuge jederzeit mit Hilfe entsprechender Befehle einerseits in den „Soft-Off“-Zustand versetzt und andererseits bei Bedarf über die Datenverbindung mit Hilfe eines „Magic-Packet“ wieder „aufgeweckt“ und in Betrieb genommen werden.

4.3.5.c Datenverbindungen via WWW-, FTP-, Telnet- und Remotedesktop

Zwischen dem PC des Operators und dem Embedded-PC des mobilen Fahrzeugs müssen via Internet Daten herunter- und hochgeladen, sowie Programme editiert, kompiliert, gestartet und beendet werden können. Auch der Embedded-PC selbst soll über das Internet administriert werden, einschließlich der o.g. Teleaktivierung und -deaktivierung. Sowohl das Betriebssystem Linux als auch Windows-XP bieten dafür die FTP-, Telnet-, Hyper-Terminal- und Remotedesktop-Verbindungen [165] [166]

[167], die auf dem Embedded-PC jeweils als entsprechender Server aktiviert sein müssen. Auch verfügt Linux ab der Erstinstallation über den Http- bzw. WWW-Server „Apache“ [168], der für Windows-XP nachträglich installiert werden muss.

Mit den Datenverbindungen FTP und Telnet können Programme und Daten übertragen (Download, Upload), mit Http, Telnet und Hyper-Terminal können Programme gestartet und mit Remotedesktop kann der Embedded-PC selbst administriert werden. Die Nutzung dieser Datenverbindungen über das Internet setzt voraus, dass das Embedded-PC-System eine feste IP-Adresse besitzt. Für weitergehende Informationen sei auf entsprechende Literatur verwiesen [169] [170].

4.3.5.d Bewegungssteuerung

Für die manuelle Steuerung, die manuelle Fernsteuerung oder die teilautonome Steuerung (s. Kap. 7.2.1) benötigt das mobile Fahrzeug entsprechende Kommandos vom Operator bzw. von der Hindernisvermeidung (Kap. 5). Die Interpretation und Umsetzung der Kommandos ist als sog. Bewegungssteuerung auf dem Mikrocontroller C167 implementiert.

Zur Kodierung der Kommandos wurden einzelne Zahlen im ASCII-Code gewählt. Im Sinne der Bedienungsfreundlichkeit wurden für die neun Bewegungsmöglichkeiten des mobilen Fahrzeugs, d.h. acht Bewegungsrichtungen und Stopp, diejenigen Zahlen von 1 bis 9 verwendet, die den Pfeilrichtungen im Nummernfeld der Computertastatur entsprechen (s. **Tab. 4.2.a - 4.2.i**).

Würde man sinnvolle Kombinationen von Bewegungsarten zulassen, dann könnten Redundanzen entfallen. Beispielsweise kann „7 = vorwärts links Bewegung“ durch „8 = vorwärts geradeaus Bewegung“ und „4 = links Lenkung“ ersetzt werden, usw. Somit reduziert sich der Umfang von Bewegungskommandos auf 2, 4, 5, 6 und 8.

Die Einstellung der gewünschten Fahrgeschwindigkeit und Lenkwinkel mit Hilfe dieser Kommandos erfolgt durch wiederholtes Senden der entsprechenden Zahl in kurzem Zeitintervall. Verlängert sich dieser Zeitintervall bzw. werden keine Kommandos gesendet, bleiben Fahrgeschwindigkeit und Lenkwinkel konstant bzw. reduzieren sich kontinuierlich. Verstreichen mehrere Sekunden nach dem letzten

Kommando, dann gilt aus Sicherheitsgründen das Kommando „5 = Stopp“ und das mobile Fahrzeug wird angehalten (s. Kap. 4.4).

- a.) 1 = rückwärts links Bewegung
- b.) 2 = rückwärts geradeaus Bewegung
- c.) 3 = rückwärts rechts Bewegung
- d.) 4 = links Lenkung (Ackermannantrieb), links Drehung (Differentialantrieb)
- e.) 5 = Stopp
- f.) 6 = rechts Lenkung (Ackermannantrieb), rechts Drehung (Differentialantrieb)
- g.) 7 = vorwärts links Bewegung
- h.) 8 = vorwärts geradeaus Bewegung
- i.) 9 = vorwärts rechts Bewegung
- j.) <http://Server-IP/cgi-bin/robotControl.exe?para1=1¶2=8>
- k.) Content-type: text/html\n\n

Tab. 4.2: a. - i.) Kommandos zur Bewegungssteuerung eines mobilen Fahrzeugs, j.) Http-Adresse für dessen Übertragung an ein CGI-Programm und k.) die Rückmeldung des CGI-Programms an den Web-Browser des Operators via Internet.

Die Kommandos in **Tab. 4.2.a - 4.2.i** kommen gleichermaßen für die manuelle Steuerung, die manuelle Fernsteuerung und die teilautonome Steuerung (s. Kap. 7.2.1) eines mobilen Fahrzeugs zum Einsatz. Sie können entweder direkt am Mikrocontroller C167 selbst, am Embedded-PC oder via Internet an die Bewegungssteuerung des Mikrocontrollers C167 eingegeben werden:

- Die einfachste Methode zur Eingabe der Kommandos in **Tab. 4.2.a - 4.2.i** an die Bewegungssteuerung kann direkt über Tasten geschehen, die an geeignete I/O-Ports mit entsprechender Programmabfrage des Mikrocontrollers C167 angeschlossen sind.
- Da der Mikrocontroller C167 über die RS232-Verbindung an den Embedded-PC gekoppelt ist, können dieselben Kommandos auch über den HyperTerminal des Embedded-PCs via RS232 an den Mikrocontrollers C167 gegeben werden.
- Auch ist es möglich, eine entsprechend programmierte Software, z.B. in C/C++, innerhalb der DOS-Eingabeaufforderung des Embedded-PC zu starten, welche die Tastatureingaben abfragt und als Kommandos via RS232 an den Mikrocontroller C167 sendet. Diese Software kann aber auch über Telnet oder Remotedesktop via Internet verwendet werden.

- Über das Internet und bei Verwendung des Http-Protokolls können die o.g. Kommandos auch mit Hilfe internetfähiger Anwendungsprogramme, wie dem Web-Browser, gesendet werden. Das Kommando wird dabei zunächst vom Web-Browser des Operator-PCs via Internet auf den Embedded-PC und dort mittels CGI (Common Gateway Interface) -Programm via RS232 an den Mikrocontroller C167 übertragen.

Dazu dient die in **Tab. 4.2.j** angegebene Http-Adresse, die als http-Link innerhalb einer WebPage oder als direkte Adresseneingabe innerhalb des Web-Browsers eingegeben und dann vom Operator-PC via Internet an den Web-Server des Embedded-PC gesendet wird. Hier müssen Protokoll, IP-Adresse des Servers, CGI-Verzeichnis, CGI-Programmname sowie der Parametername und -wert mit „?“ an die Http-Adresse angehängt werden [169] [170]. Mehrere Parameternamen und -werte werden mit „&“ getrennt. Mit Hilfe genau dieser Parameternamen und -werte können die Kommandos in **Tab. 4.2.a - 4.2.i** in eine Http-Adresse gefasst und vom Web-Browser des Operator-PCs an den Web-Server des Embedded-PCs via Internet übermittelt werden.

Zur Übergabe der Kommandos bzw. der Parameter auf dem Embedded-PC dient das in der Http-Adresse (s. **Tab. 4.2.j**) angegebene CGI-Programm „robotControl.exe“, welches im speziellen CGI-Verzeichnis des Web-Servers auf dem Embedded-PC gespeichert sein muss. Erst dann können das CGI-Programm „robotControl.exe“ via Internet gestartet, die mitgegebenen Parameter eingelesen und diese via RS232 an den Mikrocontroller C167 weitergegeben werden. Abschließend muss das CGI-Programm noch die Information „Content-type: text/html\n\n“ (**Tab. 4.2.k**) an den Web-Browser des Operators via Internet zurücksenden [169] [170].

- Mit Hilfe eines digitalen Joysticks, z.B. mit USB-, IEEE-1284-, Gameport-, Apple-Desktop-Bus-Anschluss, usw. werden die acht Bewegungsrichtungen durch Schaltkontakte erfasst. Diese können dann per C++ Programm gelesen und via Http-Verbindung an den Web-Server des Embedded-PC übertragen werden. Die Kodierung kann entsprechend **Tab. 4.2.a - 4.2.i** erfolgen, sodass das o.g. CGI-Programm „robotControl.exe“ auf dem Embedded-PC diese Daten

übernimmt und via RS232 an die Bewegungssteuerung des Mikrocontrollers C167 senden kann. Ist der Joystick mit „Force Feedback“-Funktion ausgestattet, dann könnte neben einer optischen und akustischen auch eine haptische Rückmeldung erfolgen [171] - [176]. Beispielsweise bei einer Berührung des mobilen Fahrzeugs mit einem Hindernis blockiert der Joystick für Vorwärtsbewegungen und ein akustisches Signal ertönt.

- Für die teilautonome Steuerung eines mobilen Fahrzeugs kommen die Kommandos in **Tab. 4.2.a - 4.2.i** ebenfalls zum Einsatz. Sie werden hierbei allerdings von der in Kap. 5 beschriebenen Hindernisvermeidung auf dem Embedded-PC generiert und via RS232 an die Bewegungssteuerung des Mikrocontrollers C167 übertragen (s. Kap. 7.2.1).

Die Bewegungssteuerung als solche ist in dem Software-Programm „Robot-Control“ (Kap. 4.3.6) des Mikrocontrollers C167 enthalten und generiert entsprechend den o.g. Kommandos die passenden PWM_r - und PWM_l -Signale für den Differentialantrieb bzw. PWM_v - u. PWM_ϕ -Signale für den Ackermannantrieb und gibt diese über zwei PWM-Ports des Mikrocontrollers C167 an die Motor-Control-Boards (Kap. 4.3.1) aus.

4.3.5.e Graphical User Interface (GUI)

Eine GUI stellt eine internetbasierte graphische Benutzerschnittstelle dar, die beispielsweise in Form von einer oder mehreren Web-Seiten über den Web-Server des Embedded-PC via Internet in den Web-Browser des Operators geladen werden kann. Eine GUI dient hier dem Operator zur Kontrolle des mobilen Fahrzeugs über das Internet. Zu diesem Zweck wurden hierfür drei GUI-Web-Seiten erstellt, um Videodaten der Fahrzeugumgebung zu betrachten (**Abb. 4.11**), um Steuerungsdaten zwecks Fernsteuerung zu senden (s. Kap. 4.3.5.d, **Abb. 4.12**) und um Sensor- und Posedaten zu beobachten bzw. Posedaten zu erstellen (**Abb. 4.13**) [77] [177] - [188].

Abb. 4.11 zeigt ein Beispiel einer Video-Control-GUI, mit der Videodaten via Internet betrachtet werden können. Hier wird der Fahrweg im Frontbereich des mobilen Fahrzeugs mit Hilfe einer am Chassis fixierten schwenkbaren Web-Kamera [136] gezeigt. Die in **Abb. 4.12** gezeigte Web-Seite dient als Remote-Control-GUI zur Tele-

Operation, deren Textelemente entsprechend ihrer Bedeutung jeweils mit den in **Tab. 4.2** beschriebenen Kommandos als Http-Link verknüpft sind. Diese Http-Links gestatten es, durch Anklicken mit der Computer-Maus das mobile Fahrzeug via Internet fernzusteuern.



Abb. 4.11: Video-Control-GUI zur Video-betrachtung des Fahrwegs im Frontbereich des mobilen Fahrzeugs via Internet mittels schwenkbarer Web-Kamera [136].



Abb. 4.12: Remote-Control-GUI zur Fernsteuerung eines mobilen Fahrzeugs via Internet. Die Http-Links der Textelemente sind aus **Tab. 4.2** erstellt.

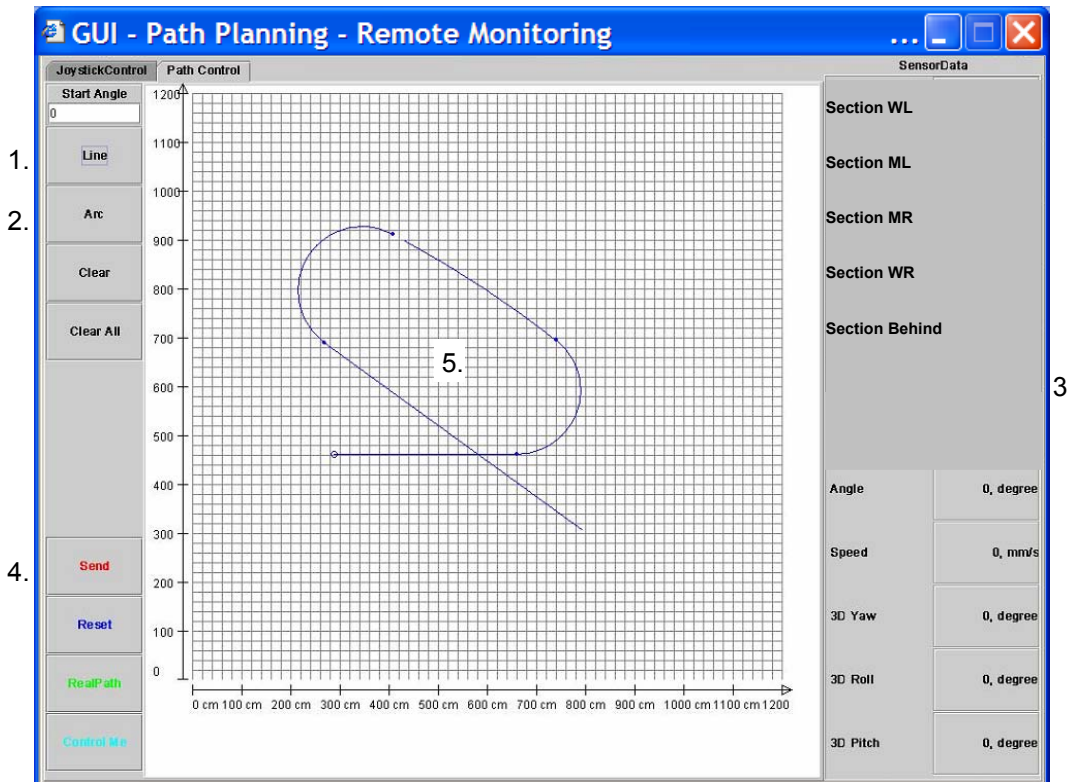


Abb. 4.13: Remote-Monitoring-GUI zur Überwachung von Sensor- und Posedaten eines mobilen Fahrzeugs via Internet sowie zur Erstellung eines Pfads [177].

(1. + 2. Auswahl von Pfadsegmenten zur Erstellung eines gewünschten Pfads, 3. Sensordaten vom und 4. Kommandos an das mobile Fahrzeug, 5. Bereich zur Erstellung eines virtuellen Pfads bzw. Visualisieren des bereits abgefahrenen Pfads).

Die Fernsteuerung eines mobilen Fahrzeugs mittels Remote-Control-GUI (**Abb. 4.12**) erfolgt zunächst ohne die Hindernis- und Fahrwegerkennung (s. Kap. 3.4) und ohne die Hindernisvermeidung (s. Kap. 5). Hier müssen alle Aktionen über die Videokontrolle (**Abb. 4.11**) durch den Operator visuell und manuell durchgeführt werden.

Bei aktivierter Hindernis- und Fahrwegerkennung sowie Hindernisvermeidung hingegen werden die Bewegungsaktionen mit Remote-Control-GUI (**Abb. 4.12**) lediglich initiiert und durch das System teilautonom ausgeführt. Fehlen Hindernisse, verhält sich das System wie eine Fernsteuerung. Beim Auftreten eines Hindernisses greift jedoch die Hindernisvermeidung ein, indem sie die in ihr implementierten Handlungen durchführt und nach deren Bewältigung zur ursprünglichen Bewegungsform zurückkehrt (s. a. Kap. 7).

4.3.6 Roboter-Controller

Der Roboter-Controller, als Software „Robot-Control.c“ in C programmiert, ist als Herzstück des mobilen Fahrzeugs zu sehen und läuft auf dem Mikrocontroller C167 als echtzeitfähiges System. Darin ist die hardware-nahe Hinderniserkennung und -vermeidung sowie die Bewegungssteuerung implementiert (s. **Abb. 4.4**). Der Roboter-Controller erfasst die Messsignale aller Sensoren des mobilen Fahrzeugs über die Eingangs-Ports bzw. Zähler des Mikrocontrollers C167. Dies sind der mechanische Front-Bumper, der IR-Front- und der IR-Boden-Bumper sowie die beiden Rad-Encoder. Über die serielle Schnittstelle RS232 werden auch die Daten der auf dem Embedded-PC implementierten Hinderniserkennung mit PMD-Kamera und Inklinationssensor empfangen.

Die Auswertung der Sensorsignale innerhalb des Roboter-Controllers erfolgt anhand der in Kap. 5 beschriebenen Hindernisvermeidung mittels Fuzzy-Logic-Regelung. Die daraus resultierenden Ausgangsgrößen bzw. Stellgrößen y_1 und y_2 werden in die PWM-Werte $PWM_l(y_1, y_2)$ und $PWM_r(y_1, y_2)$ für den Differentialantrieb bzw. $PWM_v(y_1)$ und $PWM_\phi(y_2)$ für den Ackermannantrieb konvertiert (s. Kap. 5.2.7) und an die PWM-Einheiten des Mikrocontrollers C167 ausgegeben. Als PWM-Signale im TTL-Standard gelangen diese Signale schließlich an das Motor-Control-Board MD22 [129] bzw. an den Servo-Lenkantrieb und das Motor-Control-Board MD03 [128].

Durch die Abarbeitung aller o.g. Aktionen innerhalb einer Schleifenoperation, wie Sensorsignale einlesen, auswerten und entsprechende Aktorsignale ausgeben, ergibt sich eine quasi-kontinuierliche Handlungsweise. Auch ohne Embedded-PC, also bei Hinderniserkennung ohne PMD-Kamera und ohne Inklinationssensor, lediglich ausgestattet mit dem Mikrocontroller C167 und der Roboter-Controller-Software, kann sich das mobile Fahrzeug innerhalb eines einfach strukturierten 2D-Areals teilautonom oder sogar autonom bewegen.

Mit der Ergänzung durch die PMD-Kamera und den Inklinationssensor können auch 3D-Hindernisse für die Hinderniserkennung einbezogen werden, sodass nun ebenfalls 3D-Areale mit dem mobilen Fahrzeug befahren werden können. Mit Hilfe der PMD-Kamera können darüber hinaus auch negative Hindernisse erfasst sowie der Fahrweg als solcher analysiert werden (s. Kap. 3.4).

Abb. 4.14 zeigt ein Datenflussdiagramm zur Realisierung einer (teil)autonomen Fahrbewegung eines mobilen Fahrzeugs. Darin überlagern sich zwei Prozesse. Der innere Prozess stellt die Lenk- und Geschwindigkeitsregelung dar, wobei Letztere die Rad-Encoder-Daten nutzt. Der Ackermannantrieb benötigt hierfür einen PI- oder besser einen PID-Regler für die Fahrgeschwindigkeit des Antriebsmotors und einen Servo-Regler des Servo-Lenkantriebs. Der Differentialantrieb braucht hingegen zwei identische PI- bzw. PID-Regler, jeweils einen für den linken und den rechten Antriebsmotor zwecks Regelung der Fahrgeschwindigkeit. Die Lenkung erfolgt hier durch unterschiedliche Fahrgeschwindigkeiten der Antriebsmotoren. Die Lenkregelung besteht also nur darin, dass durch zwei Sollwertvorgaben die entsprechenden Fahrgeschwindigkeiten eingeregelt werden und sich damit ein Lenkwinkel einstellt. Eine explizite Regelung des Lenkwinkels findet nicht statt. Der äußere Prozess bildet die Hindernis- und Fahrwegerkennung sowie die Hindernisvermeidung (Fuzzy-Logic-Regelung) unter Verwendung aller am mobilen Fahrzeug verbauten Sensoren einschließlich der PMD-Kamera. Alle Regler erhalten die Sollwertvorgaben von der Hindernisvermeidung (Fuzzy-Logic-Regelung).

Für den Fall, das kein Hindernis detektiert wird, fährt das mobile Fahrzeug mit konstanter vorgegebener Fahrgeschwindigkeit geradeaus. Im Falle eines Hindernisses und in Abhängigkeit von dessen Lage (Entfernung und Sichtwinkel)

relativ zum mobilen Fahrzeug werden von der Hindernisvermeidung Sollwertvorgaben generiert und an die zwei Geschwindigkeitsregler bzw. an den Geschwindigkeitsregler und Servo-Regler gegeben, um so ein Ausweichmanöver durchzuführen.

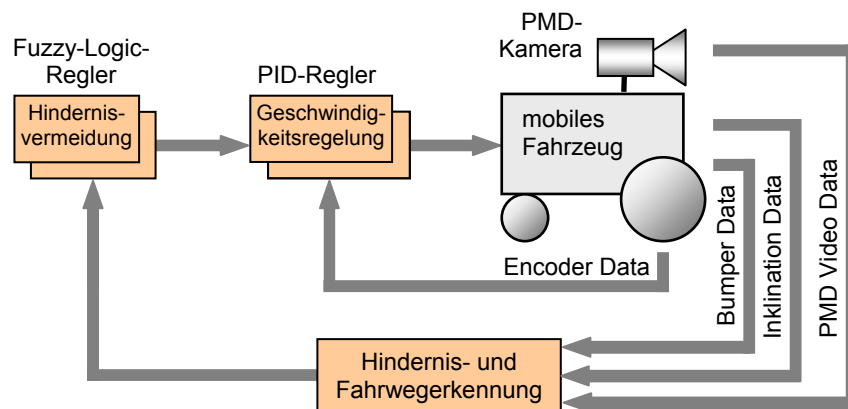


Abb. 4.14: Datenflussdiagramm zur lokalen Navigation eines mobilen Fahrzeugs bzw. fahrerlosen Transportfahrzeugs (FTF).

Da der Roboter-Controller in einer Schleifenstruktur operiert, unterteilt sich ein solches Ausweichmanöver in minimale Richtungsänderungen mit jeweils angepasster Fahrgeschwindigkeit pro Schleifendurchgang. Nach jedem Schleifendurchgang ändert sich also die Lage des Hindernisses relativ zum mobilen Fahrzeug, und zwar so lange, bis das Hindernis außerhalb des Sensorbereichs liegt und es als solches nicht mehr erfasst wird. Zusammengefasst ergeben alle minimalen Richtungsänderungen und minimalen Fahrbewegungen das gewünschte Ausweichmanöver als Bahnkurve um das Hindernis herum, ohne Kollision des mobilen Fahrzeugs mit dem Hindernis.

Damit lässt sich eine (teil)autonome Fahrbewegung ohne Umgebungskarte und ohne Pfadkontrolle innerhalb eines unstrukturierten 3D-Areals realisieren. Das mobile Fahrzeug kann für sog. ungesteuertes Fahren bzw. Zufallsfahren genutzt werden und damit ein unbekanntes Areal erkunden. Beim Durchfahren der Umgebung wird es quasi von Hindernissen „geführt“.

4.4 Sicherheitsaspekte

Mobilen Fahrzeugen begegnet man derzeit als fahrerlose Transportsysteme innerhalb von Fabrikhallen zum Transportieren z.B. von Werkstücken. Dies schließt ein, dass solche Fahrzeuge mit Personen und Objekten kollidieren können. Der

Gesetzgeber verlangt daher, dass für derartige mobile Systeme besondere Sicherheitsmaßnahmen zu treffen sind. Dazu wurden Richtlinien und Vorschriften von den Berufsgenossenschaften (BGS) [98], vom Deutschen Institut für Normung (DIN) [102] und vom Verein deutscher Ingenieure (VDI) [105] erarbeitet, die u.a. durch BGS, Technischer Überwachungsverein (TÜV) [104], Verein deutscher Sachversicherer (VdS) [109], usw. überwacht werden.

Entsprechend den BGS- [99] [100] [101], den DIN- [103] und den VDI-Vorschriften [106] [107] müssen mobile Fahrzeuge dieser Art ein Personenerkennungssystem in Form eines Auffahrschutzes, eines Bremssystems, Warneinrichtungen und Not-Aus-Einrichtungen besitzen. Für Personen sicherheitsrelevante Einrichtungen müssen unabhängig von Computer-Hardware und -Software arbeiten und direkt auf ein Not-Aus-Modul einwirken, so dass das mobile Fahrzeug unverzüglich und sicher zum Stillstand kommt [108].

Die Prozesskette von der Sensordatenerfassung mit Hilfe des Mikrocontrollers C167, der Auswertung der Sensordaten mit der entsprechenden Software bis zum Relais für die Abschaltung der Betriebsspannung der Antriebsmotoren gilt als unzuverlässig. Dies entspricht den Beanstandungen der BGS [98], des TÜV [104], des VdS [109] usw., die für Produktzertifizierungen und Betriebsgenehmigungen auch für mobile Fahrzeuge zuständig sind [3] [84]. Demnach kann die Abschaltung der Betriebsspannung entweder über ein dazwischen geschaltetes Relais oder direkt über die Schaltkontakte des jeweiligen Schaltersensors z.B. am mechanischen Front-Bumper erfolgen.

Die Abschaltung der Betriebsspannung per Relais hat den Vorteil, dass bei Hinderniskontakt nur der Vorwärtsfahrtmodus ausgeschaltet und somit ein Weiterfahren in Vorwärtsrichtung verhindert werden. Der Rückwärtsfahrtmodus hingegen soll erhalten bleiben, um so das mobile Fahrzeug in Rückwärtsrichtung wieder aus dem Hindernisbereich herausmanövrieren zu können. Die Abschaltung per Relais ist auch sinnvoll, wenn Sensoren ohne Schalterkontakt für die Hinderniserkennung verwendet werden. Bspw. können optische Distanzsensoren bei Hinderniskontakt nur ein entsprechendes Signal liefern und damit nur über ein Relais die Betriebsspannung der Antriebsmotoren ausschalten (s. Kap. 4.2). Auch können

mehrere Sensoren zwecks Hinderniserkennung mit einer Wired-OR-Verknüpfung [114] zusammengefasst werden und ein einzelnes Relais schalten (s. Kap. 4.2.2).

Für die hier beschriebenen mobilen Roboter Tom3D und Merlin3D wurden insbesondere der mechanische Front-Bumper sowie der IR-Front-Bumper (s. Kap. 4.2.1 u. 4.2.2) als Personenerkennungssystem konzipiert und mit Not-Aus-Funktion versehen. Darüber hinaus wurden Sicherheitsfunktionen bei der Unterbrechung der Kommunikation innerhalb und außerhalb des mobilen Fahrzeugs erstellt, sodass vier Sicherheitsstufen hierarchisch angeordnete werden können:

1. Sicherheitsstufe „Hinderniserkennung durch Berührung“: Der mechanische Front-Bumper schaltet bei Personen- bzw. Objektkontakt die Betriebsspannung der Antriebsmotoren direkt ab. Dies kann erreicht werden, wenn die Betriebsspannung für die Antriebsmotoren direkt über die in Reihe geschalteten Sensorschalter (Öffner) des mechanischen Front-Bumpers geführt werden. Antriebsmotoren ohne Getriebe bzw. mit Getriebe mit kleiner Untersetzung müssen auch eine Bremse pro Antriebsmotor enthalten.
2. Sicherheitsstufe „Berührungslose Hinderniserkennung“: Mit Hilfe der IR-Distanz-Sensoren des IR-Front-Bumpers kann eine kritische Distanz zu einer Person bzw. zu einem Objekt erfasst werden. Das entsprechende digitale Sensorausgangssignal schaltet direkt ein Relais, wobei ein Öffner dieses Relais innerhalb der o.g. Reihenschaltung liegen muss, um die Betriebsspannung der Antriebsmotoren auszuschalten.

Alle Schaltzustände der Sensorschalter (Öffner) und der IR-Distanz-Sensoren sollen zusätzlich im Mikrocontroller C167 erfasst und ausgewertet werden. Die Auswertung hat den Vorteil, dass ohne zusätzliche Maßnahmen die Schaltzustände aller Sensoren des mechanischen Front-Bumpers sowie des IR-Front-Bumpers jederzeit an den Operator gemeldet und ggf. auch zur Hindernisvermeidung genutzt werden können.

3. Sicherheitsstufe „Fahrzeug-Stopp bei Abbruch der RS232-Kommunikation“: Zwischen Mikrocontroller C167 und Embedded-PC besteht eine Kommunikations-

verbindung über die serielle Schnittstelle RS232. Wird diese Verbindung unterbrochen oder werden keine Daten übertragen, erfolgt ein Fahrzeug-Stopp durch die Bewegungssteuerung (s. Kap. 4.3.5.d) innerhalb des Mikrocontrollers C167.

4. Sicherheitsstufe „Fahrzeug-Stopp bei Abbruch der WLAN-, Mobilfunk- oder Satellitenkommunikation“: Wenn die Kommunikation zwischen Operator und Embedded-PC über die o.g. Wireless-Datenverbindungen abbricht. Diese Situation wird im Embedded-PC festgestellt und das Kommando „5“ entsprechen **Tab. 4.2** via RS232 an die Bewegungssteuerung innerhalb des Mikrocontrollers C167 zwecks Fahrzeug-Stopp übertragen.

5 Lokale Navigation

Teilautonomes bzw. autonomes Fahren, wie in Kap. 4 beschrieben, orientiert sich stets an lokalen Gegebenheiten der Fahrzeugumgebung. In der Fachliteratur wird dafür der bedeutungsgleiche Begriff der „lokalen Navigation“ genannt [13] [96] [189] - [192], der im Weiteren auch hier verwendet werden soll. Demnach beinhaltet die lokale Navigation sowohl die Fahrweg- und Hinderniserkennung als auch die Hindernisvermeidung.

5.1 Hardware-nahe Hinderniserkennung

Zur Detektierung von Hindernissen dienen der mechanische Front-Bumper (Kap. 4.2.1), der IR-Front-Bumper, der IR-Boden-Bumper (Kap. 4.2.2), der Inklinations-sensor (Kap. 4.2.3) und insbesondere die PMD-Kamera (s. Kap. 3.4) des mobilen Fahrzeugs. Wird nun mit Hilfe dieser Sensoren ein Hindernis erfasst, erfolgt auf der Basis zweier Fuzzy-Logic-Regler, zusammen mit den Antriebsmotoren und ihren Geschwindigkeitsreglern (PI- oder PID-Regler) eine Hindernisvermeidung. Fuzzy-Logic-Regler haben grundsätzlich den Vorteil, dass sie schnell und leicht realisiert werden können, auch ohne mathematische Kenntnis des Systemmodells.

Fahrwegerkennung, Hinderniserkennung und Hindernisvermeidung verlangen aus Sicherheitsgründen (s. Kap. 4.4) und wegen der Notwendigkeit einer kontinuierlichen Arbeitsweise eine echtzeitfähige Datenverarbeitungsumgebung. Daher werden Fuzzy-Logic-Regler, Geschwindigkeitsregler und die Zustandserfassung aller Bumper auf dem echtzeitfähigen Mikrocontroller C167 verarbeitet.

Zur Erfassung von Bilddaten mit der PMD-Kamera wird dagegen ein Embedded-PC z.B. mit WinXP als Betriebssystem benötigt. Für WinXP und Linux steht für die PMD-Kamera die notwendige Treiber- und Konfigurations-Software zur Verfügung. Darüber hinaus können externe Bildverarbeitungsalgorithmen, wie z.B. OpenCV [45] oder ICE [135], in WinXP und in Linux verwendet werden.

Alle Bumper und die PMD-Kamera sollen den Front- bzw. Bodenbereich des mobilen Fahrzeugs auf Hindernisse überwachen. Der mechanische Front-Bumper erfasst

positive Hindernisse jeweils nur im Bereich einer horizontalen Linie entlang der Stoßstange des mobilen Fahrzeugs. Der IR-Front-Bumper hingegen berücksichtigt Hindernisse, die in der Ebene vor dem mobilen Fahrzeug liegen. Aufgrund der in Bodenrichtung weisenden IR-Sensoren beim IR-Boden-Bumper können negative Hindernisse erfasst werden.

Die PMD-Kamera bildet den gesamten nahen Fahrzeugfrontbereich als PMD-Videobild ab. Aufgrund der Montagehöhe und des Neigungswinkels der PMD-Kamera können sowohl positive wie auch negative Hindernisse erfasst werden. Die PMD-Videobilder der PMD-Kamera werden stets in Kugelkoordinaten aufgenommen. Für die Weiterverarbeitung erfordert dies eine Konvertierung in kartesischen Koordinaten. Desweiteren ist der Neigungswinkel der PMD-Kamera aus jedem PMD-Videobild herauszurechnen. Bei Fahrzeugbewegungen im unebenen Gelände müssen ggf. die mit dem Inklinationsensor ermittelten Pitch- und Roll-Winkel ebenfalls verrechnet werden.

Entsprechend Kap. 3.4 erhält man so ein lageneutrales PMD-Videobild, in dem nun die Hindernisse vermessen werden können. Dazu wird das lageneutrale PMD-Videobild der PMD-Kamera in vier vertikale Segmente unterteilt und von 0 bis 3 durchnummeriert. Anschließend wird innerhalb jedes Segments die kleinste Hindernisdistanz bestimmt. Die so ermittelte Segment-Nummer $N_{Segment}$ kann auch als horizontale Hindernisposition interpretiert werden.

Diese kleinste Hindernisdistanz d_{min} zusammen mit der dazugehörenden Segment-Nummer $N_{Segment}$ bilden die Basisvariablen zweier relationalen Fuzzy-Logic-Regler, die als Ausgangsgrößen jeweils die Fahrgeschwindigkeit und den Lenkwinkel ermitteln und ausgeben sollen.

5.2 Hindernisvermeidung mit Fuzzy-Logic-Regelung

Ein mobiles Fahrzeug bewegt sich hier nun auf einer ebenen Fläche, mit leichten Unebenheiten. Als Grundgedanke bei der Hinderniserkennung interessiert nun, an welcher Position im horizontalen Blickfeld sich das am Nächsten gelegene Hindernis befindet und wie weit dieses vom mobilen Fahrzeug entfernt liegt. Die Hindernis-

vermeidung muss entsprechend darauf reagieren. Dies kann ein einfaches Anhalten, ein Richtungswechsel, ein Ausweichmanöver mit neuer Pose neben dem Hindernis oder ein Umfahrungsmanöver mit Rückkehr zum alten Pfad bedeuten:

1. Im einfachsten Fall einer Hindernisvermeidung wird die Fahrbewegung des mobilen Fahrzeugs komplett unterbrochen (s. **Abb. 5.1.a**). Diese Art der Hindernisvermeidung wird beispielsweise bei FTF in Produktionsbetrieben angewandt, die aus Sicherheitsgründen die vorgegebene Fahrbahn nicht verlassen dürfen [3] [84] [99] - [109].
2. Eine Hindernisvermeidung durch einfache Richtungswechsel (s. **Abb. 5.1.b**) vollzieht sich wie folgt: Wenn ein Hindernis auf **weit links**, **mitte links**, **mitte rechts** oder **weit rechts** im Frontbereich des mobilen Fahrzeugs detektiert wird, muss das mobile Fahrzeug eine Lenkrichtung nach **leicht rechts**, **weit rechts**, **weit links** oder **leicht links** mit entsprechendem Lenkwinkel, wie z.B. $-22,25^\circ$, -45° , $+45^\circ$, $+22,25^\circ$, einschlagen und eine Kurve fahren. Anschließend soll das mobile Fahrzeug geradeaus weiterfahren. Bei Hindernissen, die zu nahe am mobilen Fahrzeug liegen, muss ggf. eine Rückwärtsfahrt durchgeführt werden, um beim anschließenden Lenkvorgang keinen mechanischen Kontakt des mobilen Fahrzeugs mit dem Hindernis zu provozieren.

Solche Richtungswechsel sind z.B. bei der Exploration eines unbekanntes Areal sinnvoll, bei der das mobile Fahrzeug ohne Umgebungskarte und ohne vorgegebenen Pfad mit hoher Bewegungsfreiheit operieren soll. Beispielsweise agieren so Rasenmäher- oder Staubsaugerroboter, die ohne Umgebungskarte von Hindernissen „geführt“ sich mittels solcher Richtungswechsel zufällig durch das Areal bewegen.

3. Eine Hindernisvermeidung, bei der einem Hindernis ausgewichen werden soll, wird ähnlich wie in 2. erzielt, jedoch mit kleineren Lenkwinkeln sowie anschließender Zielausrichtung und -fahrt. Wird nach der letzten Lenk- / Fahraktion kein Hindernis mehr detektiert, erfolgt eine Ausrichtung des mobilen Fahrzeugs auf den ursprünglichen Zielpunkt. Erst wenn dabei kein Hindernis entdeckt wird, soll das mobile Fahrzeug auf den Zielpunkt zufahren (s. **Abb. 5.1.c**).

Letztere Aktion erfordert entsprechend Kap. 6 die Kenntnis der eigenen Pose und die des Zielpunktes sowie entsprechend Kap. 7.2.3 die Fähigkeiten eigenständig eine „Orientierung einzunehmen“ und eine „Pose anzufahren“.

Der Lenkvorgang muss dabei rechtzeitig vor dem Hindernis bei verlangsamer Fahrt beginnen. Während des Lenk- / Fahrmanövers, d.h. fortlaufend Hindernis detektieren, lenken und kleine Distanz fahren, ändert sich stets die Lage des Hindernisses relativ zum mobilen Fahrzeug, und zwar so lange, bis das Hindernis außerhalb des Sensorbereichs liegt. Das mobile Fahrzeug muss dabei seinen alten Pfad verlassen, eine Kurve um das Hindernis fahren und danach eine neue Pose neben dem Hindernis einnehmen (s. **Abb. 5.1.c**). Beispielsweise bei Transportaufgaben und Botengängen mit oder ohne vorgegebenen Pfad kann das mobile Fahrzeug nach Abschluss des Ausweichmanövers von dieser neuen Pose aus direkt auf dem kürzesten Weg zum Zielpunkt weiterfahren.

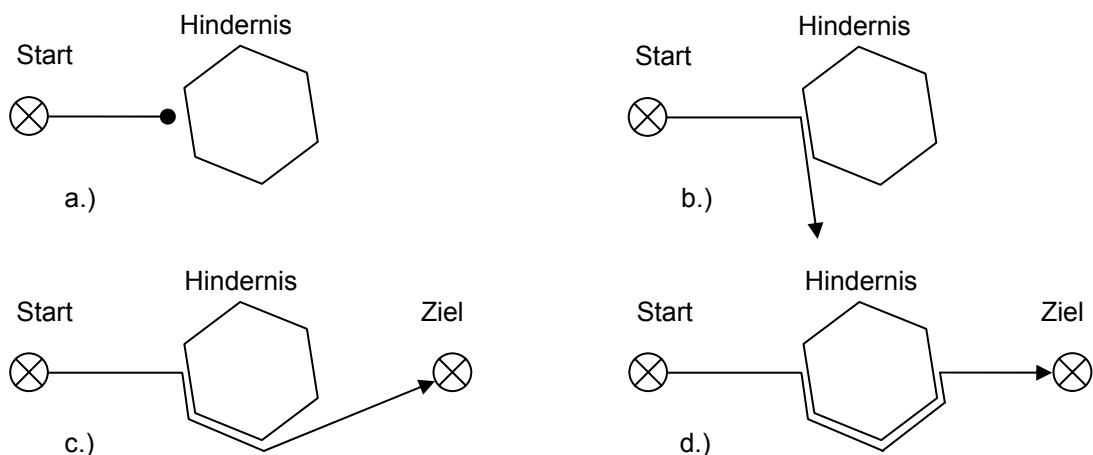


Abb. 5.1: Varianten der Hindernisvermeidung: a.) Stopp, b.) Richtungsänderung, c.) Ausweichmanöver und d.) Umfahrungsmanöver mit Rückkehr zum alten Kurs.

4. Für das Umfahrungsmanöver benötigt das mobile Fahrzeug die Fähigkeit, entlang einer Wand mit definiertem Abstand zu fahren. Im Falle eines Hindernisses soll das mobile Fahrzeug entlang dessen Kontur fahren (s. **Abb. 5.1.d**). Als Abbruchkriterium dient nun die Situation, in der das mobile Fahrzeug mit seiner Ist-Position den geplanten Pfad kreuzt. Andernfalls würde das mobile Fahrzeug permanent das Hindernis umrunden.

Das mobile Fahrzeug kehrt damit, nach der Umfahrung hinter dem Hindernis, zum alten Pfad zurück. Das Umfahrungsmanöver setzt voraus, dass neben

einem Start- und Zielpunkt auch ein geplanter Pfad dazwischen vorhanden ist. Des Weiteren ist sowohl für das Pfadfahren selbst als auch für das Wiederfinden des Pfads eine Selbstlokalisierung notwendig (s. Kap. 6).

Damit kann das mobile Fahrzeug der Kontur des Hindernisses folgen und zwar so lange, bis sich die aktuelle Position mit dem geplanten Pfad hinter dem Hindernis überlagert. Das Hindernis wurde somit umfahren und das mobile Fahrzeug kann jetzt von dieser neuen Position aus wieder auf dem geplanten Pfad weiterfahren. Beispielsweise soll ein mobiles Fahrzeug die Dienstleistung Rasenmähen, Reinigen, Kehren, usw. ausschließlich auf vorgegebenen Bahnen durchführen und bei einer evtl. Hindernisumfahrung genau hinter dem Hindernis seine Tätigkeit wieder fortsetzen.

Das Ausweichmanöver, welches oben unter 2. beschrieben und in **Abb. 5.1.b** gezeigt ist, soll hier nun mit Hilfe zweier relationaler Fuzzy-Logic-Regler realisiert werden. Ein Fuzzy-Logic-Regler unterteilt sich in die drei Komponenten Fuzzyfizierung, Fuzzy-Inferenz und Defuzzyfizierung (**Abb. 5.2**). Die Hindernisdistanz d_{min} sowie die Segment-Nummer $N_{Segment}$ bilden die Basisvariablen. Die Fahrgeschwindigkeit sowie der Lenkwinkel stellen die Ausgangsgrößen der Fuzzy-Logic-Regler dar.

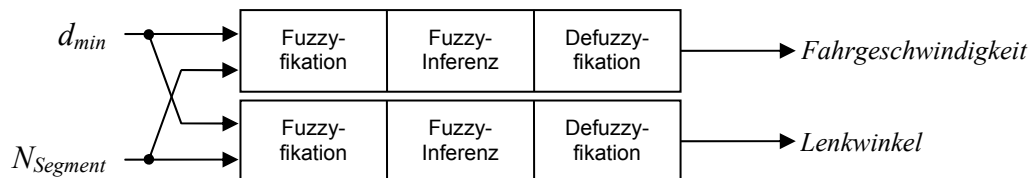


Abb. 5.2: Zwei Relationale Fuzzy-Logic-Regler, jeweils unterteilt in Fuzzyfizierung, Fuzzy-Inferenz und Defuzzyfizierung [130] [131].

5.2.1 Fuzzyfizierung

Zunächst wird das mit den Bumpern oder der PMD-Kamera erfasste Hindernis als ein weit linkes (WL), ein mittel linkes (ML), ein mittlerechtes (MR) oder ein weit rechtes (WR) Hindernis im Frontbereich des mobilen Fahrzeugs interpretiert. Des Weiteren muss die mit der PMD-Kamera gemessene minimale Hindernisdistanz als eine unterschrittene Distanz (UD) mit $< 1,5$ m, eine Grenzdistanz (GD) mit ca. 1,5 m, eine nahe Distanz (ND) mit $> 1,5$ bis 3 m, eine mittlere Distanz (MD) mit $> 2,5$ bis 4 m oder eine ferne Distanz (FD) mit $> 3,5$ m eingestuft werden.

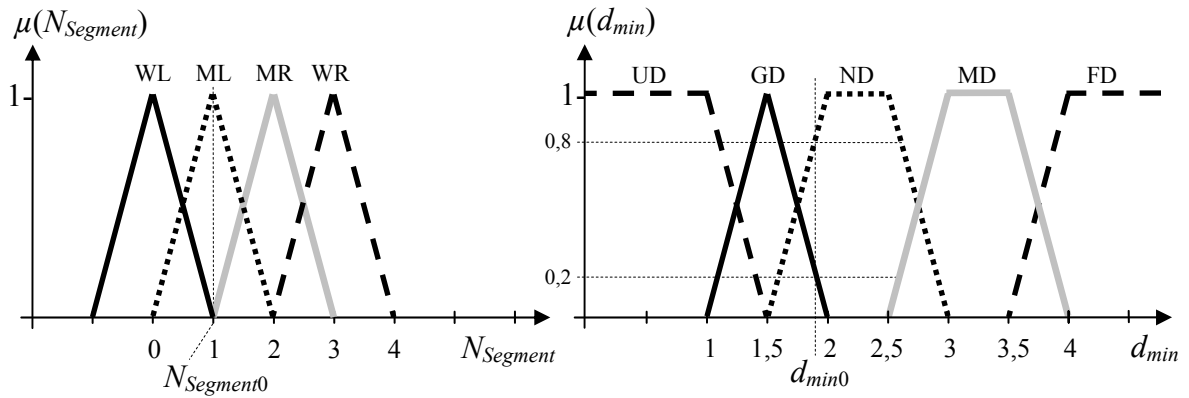


Abb. 5.3: Linguistische Werte (WL, ML, MR, WR und UD, GD, ND, MD, FD) der linguistischen Variablen „horizontale Hindernisposition“ (l.) und „Hindernisdistanz“ (r.) mit ihren Basisvariablen „ $N_{Segment}$ “ und „ d_{min} “.

$$\mu_{\Delta}(x) = \begin{cases} 0 & \text{für } x < x_{\Delta a} \\ (x - x_{\Delta a}) / (x_{\Delta m} - x_{\Delta a}) & \text{für } x_{\Delta a} \leq x \leq x_{\Delta m} \\ (x_{\Delta e} - x) / (x_{\Delta e} - x_{\Delta m}) & \text{für } x_{\Delta m} \leq x \leq x_{\Delta e} \\ 0 & \text{für } x > x_{\Delta e} \end{cases} \quad (5.1)$$

$$\mu_{\square}(x) = \begin{cases} 0 & \text{für } x < x_{\square a} \\ (x - x_{\square a}) / (x_{\square l} - x_{\square a}) & \text{für } x_{\square a} \leq x \leq x_{\square l} \\ 1 & \text{für } x_{\square l} \leq x \leq x_{\square r} \\ (x_{\square e} - x) / (x_{\square e} - x_{\square r}) & \text{für } x_{\square r} \leq x \leq x_{\square e} \\ 0 & \text{für } x > x_{\square e} \end{cases} \quad (5.2)$$

	$x_{\Delta a}$	$x_{\Delta m}$	$x_{\Delta e}$
$\mu_{WL}(N_{Segment})$	-1	0	1
$\mu_{ML}(N_{Segment})$	0	1	2
$\mu_{MR}(N_{Segment})$	1	2	3
$\mu_{WR}(N_{Segment})$	2	3	4

Tab. 5.1: Datenbasis zur Fuzzyfizierung der Basisvariablen „ $N_{Segment}$ “ der linguistischen Variablen „horizontale Hindernisposition“.

	$x_{\Delta a}$	$x_{\Delta m}$	$x_{\Delta e}$	$x_{\square a}$	$x_{\square l}$	$x_{\square r}$	$x_{\square e}$
$\mu_{UD}(d_{min})$				$-\infty$	$-\infty$	1	1,5
$\mu_{GD}(d_{min})$	1	1,5	2				
$\mu_{ND}(d_{min})$				1,5	2	2,5	3
$\mu_{MD}(d_{min})$				2,5	3	3,5	4
$\mu_{FD}(d_{min})$				3,5	4	∞	∞

Tab. 5.2: Datenbasis zur Fuzzyfizierung der Basisvariablen „ d_{min} “ der linguistischen Variablen „Hindernisdistanz“.

Die Abkürzungen WL, ML, MR, WR und UD, GD, ND, MD, FD stellen linguistische Werte der linguistischen Variablen „horizontale Hindernisposition“ und „Hindernisdistanz“ dar. Deren jeweiligen Zugehörigkeitsfunktionen werden mit Hilfe von

Dreiecks- und Trapezfunktionen dargestellt (s. **Abb. 5.3**), die mittels der Gleichungen (5.1) (5.2) zu berechnen sowie in den **Tab. 5.1** und **5.2** formuliert sind.

Bei der Fuzzyfizierung werden nun die „scharfen“ Eingangsgrößen der Basisvariablen $N_{Segment}$ (Segment-Nummer) und d_{min} (minimale Distanz) anhand der Zugehörigkeitsfunktion in linguistische Werte (WL, ML, MR, WR und UD, GD, ND, MD, FD) umgewandelt (s. **Abb. 5.3**). Für jeden linguistischen Wert wird so der Zugehörigkeitsgrad ermittelt und dementsprechend als Vektor zusammengefasst.

Als Ergebnis der Fuzzyfizierung erhält man die fuzzyfizierten Eingangswerte $N_{SegmentF}$ und d_{minF} jeweils in Vektorform. Mit $N_{Segment0} = 1$ und $d_{min0} = 1,85$ m (s. a. **Abb. 5.3**) findet sich in den Gleichungen (5.3a) und (5.3b) ein konkretes Beispiel:

$$N_{SegmentF}(N_{Segment}) = [\mu_{WL}(N_{Segment}), \mu_{ML}(N_{Segment}), \mu_{MR}(N_{Segment}), \mu_{WR}(N_{Segment})] \quad (5.3a)$$

$$N_{SegmentF}(N_{Segment0}) = [0, 1, 0, 0]$$

$$d_{minF}(d_{min}) = [\mu_{UD}(d_{min}), \mu_{GD}(d_{min}), \mu_{ND}(d_{min}), \mu_{MD}(d_{min}), \mu_{FD}(d_{min})] \quad (5.3b)$$

$$d_{minF}(d_{min0}) = [0, 0,2, 0,8, 0, 0]$$

5.2.2 Fuzzy-Inferenz: Regelbasis

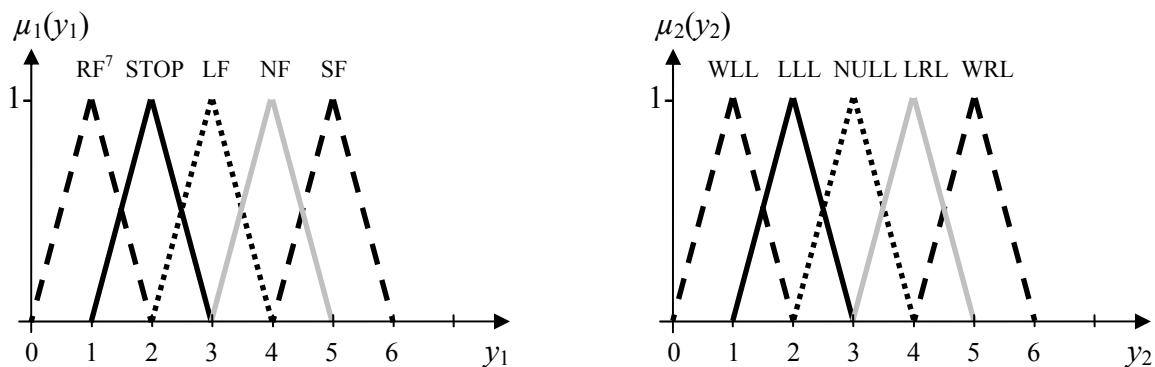


Abb. 5.4: Linguistische Werte der Stellgrößen „Geschwindigkeit“ y_1 , „Lenkwinkel“ y_2 (RF⁷ = rückwärts fahren, STOP = Stopp, LF = langsam vorwärts fahren, NF = normal vorwärts fahren, SF= schnell vorwärts fahren, WLL = weit links lenken, LLL = leicht links lenken, NULL = nicht lenken, LRL = leicht rechts lenken, WRL = weit rechts lenken).

$$\mu_{\Delta}(y) = \begin{cases} 0 & \text{für } y < y_{\Delta a} \\ (y - y_{\Delta a}) / (y_{\Delta m} - y_{\Delta a}) & \text{für } y_{\Delta a} \leq y \leq y_{\Delta m} \\ (y_{\Delta e} - y) / (y_{\Delta e} - y_{\Delta m}) & \text{für } y_{\Delta m} \leq y \leq y_{\Delta e} \\ 0 & \text{für } y > y_{\Delta e} \end{cases} \quad (5.4)$$

⁷ Rückwärts fahren impliziert zunächst einen Fahrstopp, dann ca. 40 cm rückwärts fahren und schließlich wieder einen Fahrstopp.

Neben den in **Abb. 5.3** dargestellten Zugehörigkeitsfunktionen der linguistischen Variablen „horizontale Hindernisposition“ und „Hindernisdistanz“ erfordert ein Fuzzy-Regler auch Zugehörigkeitsfunktionen der linguistischen Werte für die Ausgangsvariablen bzw. Stellgrößen „Geschwindigkeit“ y_1 und „Lenkwinkel“ y_2 (**Abb. 5.4**). Die Stellgrößen y_1 und y_2 sind als Reaktion auf die Eingangswerte zu sehen, die bei der Fahrbewegung des mobilen Fahrzeugs durchzuführen sind. Die Verknüpfung der Eingangswerte mit den Ausgangswerten mittels IF-THEN-Beziehungen repräsentiert die gewünschte Hindernisvermeidung.

	$y_{\Delta a}$	$y_{\Delta m}$	$y_{\Delta e}$
$\mu_{RF}(y_1)$	0	1	2
$\mu_{STOP}(y_1)$	1	2	3
$\mu_{LF}(y_1)$	2	3	4
$\mu_{NF}(y_1)$	3	4	5
$\mu_{SF}(y_1)$	4	5	6

	$y_{\Delta a}$	$y_{\Delta m}$	$y_{\Delta e}$
$\mu_{WLL}(y_2)$	0	1	2
$\mu_{LLL}(y_2)$	1	2	3
$\mu_{NULL}(y_2)$	2	3	4
$\mu_{LRL}(y_2)$	3	4	5
$\mu_{WRL}(y_2)$	4	5	6

Tab. 5.3: Datenbasis zur Geschwindigkeit y_1 . **Tab. 5.4:** Datenbasis zum Lenkwinkel y_2 .

R ₀₁ : IF $N_{Segment}=WL$ AND $d_{min}=UD$ OR $N_{Segment}=ML$ AND $d_{min}=UD$ OR $N_{Segment}=MR$ AND $d_{min}=UD$ OR $N_{Segment}=WR$ AND $d_{min}=UD$ THEN $y_1=RF^7$	R ₀₈ : IF $N_{Segment}=WL$ AND $d_{min}=UD$ OR $N_{Segment}=ML$ AND $d_{min}=UD$ OR $N_{Segment}=MR$ AND $d_{min}=UD$ OR $N_{Segment}=WR$ AND $d_{min}=UD$ THEN $y_2=NULL$
R ₀₂ : IF $N_{Segment}=WL$ AND $d_{min}=GD$ OR $N_{Segment}=ML$ AND $d_{min}=GD$ OR $N_{Segment}=MR$ AND $d_{min}=GD$ OR $N_{Segment}=WR$ AND $d_{min}=GD$ THEN $y_1=STOP$	R ₀₉ : IF $N_{Segment}=WL$ AND $d_{min}=GD$ OR $N_{Segment}=ML$ AND $d_{min}=GD$ THEN $y_2=WRL$
R ₀₃ : IF $N_{Segment}=ML$ AND $d_{min}=ND$ OR $N_{Segment}=MR$ AND $d_{min}=ND$ THEN $y_1=LF$	R ₁₀ : IF $N_{Segment}=MR$ AND $d_{min}=GD$ OR $N_{Segment}=WR$ AND $d_{min}=GD$ THEN $y_2=WLL$
R ₀₄ : IF $N_{Segment}=WL$ AND $d_{min}=ND$ OR $N_{Segment}=WR$ AND $d_{min}=ND$ THEN $y_1=NF$	R ₁₁ : IF $N_{Segment}=WL$ AND $d_{min}=ND$ THEN $y_2=LRL$
R ₀₅ : IF $N_{Segment}=ML$ AND $d_{min}=MD$ OR $N_{Segment}=MR$ AND $d_{min}=MD$ THEN $y_1=NF$	R ₁₂ : IF $N_{Segment}=ML$ AND $d_{min}=ND$ THEN $y_2=WRL$
R ₀₆ : IF $N_{Segment}=WL$ AND $d_{min}=MD$ OR $N_{Segment}=WR$ AND $d_{min}=MD$ THEN $y_1=SF$	R ₁₃ : IF $N_{Segment}=MR$ AND $d_{min}=ND$ THEN $y_2=WLL$
R ₀₇ : IF $N_{Segment}=WL$ AND $d_{min}=FD$ OR $N_{Segment}=ML$ AND $d_{min}=FD$ OR $N_{Segment}=MR$ AND $d_{min}=FD$ OR $N_{Segment}=WR$ AND $d_{min}=FD$ THEN $y_1=SF$	R ₁₄ : IF $N_{Segment}=WR$ AND $d_{min}=ND$ THEN $y_2=LLL$
	R ₁₅ : IF $N_{Segment}=ML$ AND $d_{min}=MD$ THEN $y_2=LRL$
	R ₁₆ : IF $N_{Segment}=MR$ AND $d_{min}=MD$ THEN $y_2=LLL$
	R ₁₇ : IF $N_{Segment}=WL$ AND $d_{min}=MD$ OR $N_{Segment}=WR$ AND $d_{min}=MD$ THEN $y_2=NULL$
	R ₁₈ : IF $N_{Segment}=WL$ AND $d_{min}=FD$ OR $N_{Segment}=ML$ AND $d_{min}=FD$ OR $N_{Segment}=MR$ AND $d_{min}=FD$ OR $N_{Segment}=WR$ AND $d_{min}=FD$ THEN $y_2=NULL$

Tab. 5.5: Inferenz-Regeln R₀₁ - R₁₈.

		$N_{Segment}$			
		WL	ML	MR	WR
d_{min}	UD	RF ⁷	RF ⁷	RF ⁷	RF ⁷
	GD	STOP	STOP	STOP	STOP
	ND	NF	LF	LF	NF
	MD	SF	NF	NF	SF
	FD	SF	SF	SF	SF

Tab. 5.6: Regelbasis in Tabellenform für die Geschwindigkeit y_1 .

		$N_{Segment}$			
		WL	ML	MR	WR
d_{min}	UD	NULL	NULL	NULL	NULL
	GD	WRL	WRL	WLL	WLL
	ND	LRL	WRL	WLL	LLL
	MD	NULL	LRL	LLL	NULL
	FD	NULL	NULL	NULL	NULL

Tab. 5.7: Regelbasis in Tabellenform für den Lenkwinkel y_2 .

5.2.3 Fuzzy-Inferenz: Auswertung der Regelprämissen

Wenn die Prämisse in elementarer Form vorliegt, es sich somit um eine unscharfe einstellige Relation handelt, dann entspricht der Erfüllungsgrad α genau dem bei der Fuzzifizierung ermittelten Zugehörigkeitswert [132]. Bei mehrstelliger Relation erfolgt eine sog. Auswertung der Regelprämisse, wobei für die unscharfe AND-Verknüpfung der MIN-Operator (t-Norm-Operator) und für die unscharfe OR-Verknüpfung der MAX-Operator (t-Konorm-Operator) zur Anwendung kommt [130] [131]. Aus den in den **Tabellen 5.5, 5.6** und **5.7** aufgestellten Inferenz-Regeln lassen sich die Erfüllungs- bzw. Aktivierungsgrade α wie folgt berechnen:

$$\alpha_{RF} = \max[\min(\mu_{WL}(N_{Segment}), \mu_{UD}(d_{min})), \min(\mu_{ML}(N_{Segment}), \mu_{UD}(d_{min})), \min(\mu_{MR}(N_{Segment}), \mu_{UD}(d_{min})), \min(\mu_{WR}(N_{Segment}), \mu_{UD}(d_{min}))] \quad (5.5a)$$

$$\alpha_{STOP} = \max[\min(\mu_{WL}(N_{Segment}), \mu_{GD}(d_{min})), \min(\mu_{ML}(N_{Segment}), \mu_{GD}(d_{min})), \min(\mu_{MR}(N_{Segment}), \mu_{GD}(d_{min})), \min(\mu_{WR}(N_{Segment}), \mu_{GD}(d_{min}))] \quad (5.5b)$$

$$\alpha_{LF} = \max[\min(\mu_{ML}(N_{Segment}), \mu_{ND}(d_{min})), \min(\mu_{MR}(N_{Segment}), \mu_{ND}(d_{min}))] \quad (5.5c)$$

$$\alpha_{NF} = \max[\min(\mu_{WL}(N_{Segment}), \mu_{ND}(d_{min})), \min(\mu_{ML}(N_{Segment}), \mu_{MD}(d_{min})), \min(\mu_{MR}(N_{Segment}), \mu_{MD}(d_{min})), \min(\mu_{WR}(N_{Segment}), \mu_{ND}(d_{min}))] \quad (5.5d)$$

$$\alpha_{SF} = \max[\min(\mu_{WL}(N_{Segment}), \mu_{MD}(d_{min})), \min(\mu_{WR}(N_{Segment}), \mu_{MD}(d_{min})), \min(\mu_{WL}(N_{Segment}), \mu_{FD}(d_{min})), \min(\mu_{ML}(N_{Segment}), \mu_{FD}(d_{min})), \min(\mu_{MR}(N_{Segment}), \mu_{FD}(d_{min})), \min(\mu_{WR}(N_{Segment}), \mu_{FD}(d_{min}))] \quad (5.5e)$$

$$\alpha_{WLL} = \max[\min(\mu_{MR}(N_{Segment}), \mu_{GD}(d_{min})), \min(\mu_{MR}(N_{Segment}), \mu_{ND}(d_{min})), \min(\mu_{WR}(N_{Segment}), \mu_{GD}(d_{min}))] \quad (5.5f)$$

$$\alpha_{LLL} = \max[\min(\mu_{MR}(N_{Segment}), \mu_{MD}(d_{min})), \min(\mu_{WR}(N_{Segment}), \mu_{ND}(d_{min}))] \quad (5.5g)$$

$$\alpha_{LRL} = \max[\min(\mu_{WL}(N_{Segment}), \mu_{ND}(d_{min})), \min(\mu_{ML}(N_{Segment}), \mu_{MD}(d_{min}))] \quad (5.5h)$$

$$\alpha_{\text{NULL}} = \max[\min(\mu_{\text{WL}}(N_{\text{Segment}}), \mu_{\text{UD}}(d_{\text{min}})), \min(\mu_{\text{ML}}(N_{\text{Segment}}), \mu_{\text{UD}}(d_{\text{min}})), \\ \min(\mu_{\text{MR}}(N_{\text{Segment}}), \mu_{\text{UD}}(d_{\text{min}})), \min(\mu_{\text{WR}}(N_{\text{Segment}}), \mu_{\text{UD}}(d_{\text{min}})), \\ \min(\mu_{\text{WL}}(N_{\text{Segment}}), \mu_{\text{MD}}(d_{\text{min}})), \min(\mu_{\text{WR}}(N_{\text{Segment}}), \mu_{\text{MD}}(d_{\text{min}})), \\ \min(\mu_{\text{WL}}(N_{\text{Segment}}), \mu_{\text{FD}}(d_{\text{min}})), \min(\mu_{\text{ML}}(N_{\text{Segment}}), \mu_{\text{FD}}(d_{\text{min}})), \\ \min(\mu_{\text{MR}}(N_{\text{Segment}}), \mu_{\text{FD}}(d_{\text{min}})), \min(\mu_{\text{WR}}(N_{\text{Segment}}), \mu_{\text{FD}}(d_{\text{min}}))] \quad (5.5i)$$

$$\alpha_{\text{WRL}} = \max[\min(\mu_{\text{WL}}(N_{\text{Segment}}), \mu_{\text{GD}}(d_{\text{min}})), \min(\mu_{\text{ML}}(N_{\text{Segment}}), \mu_{\text{GD}}(d_{\text{min}})), \\ \min(\mu_{\text{ML}}(N_{\text{Segment}}), \mu_{\text{ND}}(d_{\text{min}}))] \quad (5.5j)$$

Die Gesamterfüllungsgrade α_1 und α_2 im Allgemeinen und zusätzlich als konkretes Beispiel mit $N_{\text{Segment}0}=1$ und $d_{\text{min}0}=1,85\text{m}$ (s. a. **Abb. 5.3**) sind in (5.6) dargestellt:

$$\alpha_1 = [\alpha_{\text{RF}}, \alpha_{\text{STOP}}, \alpha_{\text{LF}}, \alpha_{\text{NF}}, \alpha_{\text{SF}}] \quad (5.6a)$$

$$\alpha_1 = [0, 0,2, 0,8, 0, 0]$$

$$\alpha_2 = [\alpha_{\text{WLL}}, \alpha_{\text{LLL}}, \alpha_{\text{NULL}}, \alpha_{\text{LRL}}, \alpha_{\text{WRL}}] \quad (5.6b)$$

$$\alpha_2 = [0, 0, 0, 0, 0,8]$$

5.2.4 Fuzzy-Inferenz: Regelaktivierung

Die Erfüllungsgrade, auch Aktivierungsgrade genannt, in (5.5) aktivieren jeweils die entsprechenden Zugehörigkeitsfunktion RF, STOP, LF, NF, SF sowie WLL, LLL, NULL, LRL, WRL in **Abb. 5.4**. Die Aktivierung selbst erfolgt mit Hilfe der sog. MIN-Methode nach Mamdani entsprechend Gleichung (5.7a) und (5.8a) [199]:

$$\mu_{i\text{min}}(y_1) = \min(\alpha_i, \mu_i(y_1)) \quad \text{mit } i = \text{"RF", "STOP", "LF", "NF", "SF"} \quad (5.7a)$$

$$\mu_{1\text{min}}(y_1) = [\mu_{\text{RFmin}}(y_1), \mu_{\text{STOPmin}}(y_1), \mu_{\text{LFmin}}(y_1), \mu_{\text{NFmin}}(y_1), \mu_{\text{SFmin}}(y_1)] \quad (5.7b)$$

$$= [\min(\alpha_{\text{RF}}, \mu_{\text{RF}}(y_1)), \min(\alpha_{\text{STOP}}, \mu_{\text{STOP}}(y_1)), \min(\alpha_{\text{LF}}, \mu_{\text{LF}}(y_1)), \\ \min(\alpha_{\text{NF}}, \mu_{\text{NF}}(y_1)), \min(\alpha_{\text{SF}}, \mu_{\text{SF}}(y_1))] \\ = [\min(0, \mu_{\text{RF}}(y_1)), \min(0,2, \mu_{\text{STOP}}(y_1)), \\ \min(0,8, \mu_{\text{LF}}(y_1)), \min(0, \mu_{\text{NF}}(y_1)), \min(0, \mu_{\text{SF}}(y_1))]]$$

$$\mu_{j\text{min}}(y_2) = \min(\alpha_j, \mu_j(y_2)) \quad \text{mit } j = \text{"WLL", "LLL", "NULL", "LRL", "WRL"} \quad (5.8a)$$

$$\mu_{2\text{min}}(y_2) = [\mu_{\text{WLLmin}}(y_2), \mu_{\text{LLLmin}}(y_2), \mu_{\text{NULLmin}}(y_2), \mu_{\text{LRLmin}}(y_2), \mu_{\text{WRLmin}}(y_2)] \quad (5.8b)$$

$$= [\min(\alpha_{\text{WLL}}, \mu_{\text{WLL}}(y_2)), \min(\alpha_{\text{LLL}}, \mu_{\text{LLL}}(y_2)), \\ \min(\alpha_{\text{NULL}}, \mu_{\text{NULL}}(y_2)), \min(\alpha_{\text{LRL}}, \mu_{\text{LRL}}(y_2)), \min(\alpha_{\text{WRL}}, \mu_{\text{WRL}}(y_2))] \\ = [\min(0, \mu_{\text{WLL}}(y_2)), \min(0, \mu_{\text{LLL}}(y_2)), \min(0, \mu_{\text{NULL}}(y_2)), \\ \min(0, \mu_{\text{LRL}}(y_2)), \min(0,8, \mu_{\text{WRL}}(y_2))]]$$

Als Ergebnis der MIN-Methode nach Mamdani erhält man mit den Gleichungen (5.7) und (5.8) die in **Abb. 5.5** gezeigten aktivierten Zugehörigkeitsfunktionen. Neben der MIN-Methode ist in der Regelungstechnik auch die sog. PROD-Methode

entsprechend Gleichung (5.9) gebräuchlich, in der die jeweilige Zugehörigkeitsfunktion mit dem entsprechenden Erfüllungs- bzw. Aktivierungsgrad multipliziert wird.

$$\mu_{i\text{prod}}(y_1) = \alpha_i \cdot \mu_i(y_1) \quad \text{mit} \quad i = \text{"RF", "STOP", "LF", "NF", "SF"} \quad (5.9a)$$

$$\mu_{j\text{prod}}(y_2) = \alpha_j \cdot \mu_j(y_2) \quad \text{mit} \quad j = \text{"WLL", "LLL", "NULL", "LRL", "WRL"} \quad (5.9b)$$

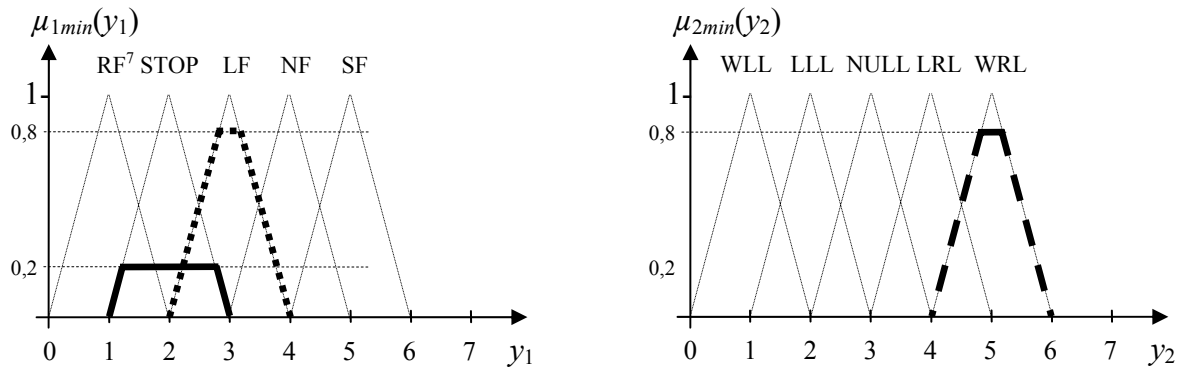


Abb. 5.5: Aktivierte Zugehörigkeitsfunktionen mit Hilfe der MIN-Methode nach Mamdani für die Geschwindigkeit (l.) und für den Lenkwinkel (r.).

5.2.5 Fuzzy-Inferenz: Aggregation

Die Aggregation fasst die aus **Abb. 5.5** resultierenden Zugehörigkeitsfunktionen zu einer gesamten Funktion zusammen (s. **Abb. 5.6**). Dabei kommt die MAX-MIN-Inferenz-Methode entsprechend (5.10) und (5.11) zur Anwendung:

$$\mu_{i\text{min}}(y_1) = \min(\alpha_i, \mu_i(y_1)) \quad \text{mit} \quad i = \text{"RF", "STOP", "LF", "NF", "SF"} \quad (5.10a)$$

$$\begin{aligned} \mu_{1\text{maxmin}}(y_1) &= \max [\mu_{\text{RFmin}}(y_1), \mu_{\text{STOPmin}}(y_1), \mu_{\text{LFmin}}(y_1), \mu_{\text{NFmin}}(y_1), \mu_{\text{SFmin}}(y_1)] \\ &= \max [\min(\alpha_{\text{RF}}, \mu_{\text{RF}}(y_1)), \min(\alpha_{\text{STOP}}, \mu_{\text{STOP}}(y_1)), \min(\alpha_{\text{LF}}, \mu_{\text{LF}}(y_1)), \\ &\quad \min(\alpha_{\text{NF}}, \mu_{\text{NF}}(y_1)), \min(\alpha_{\text{SF}}, \mu_{\text{SF}}(y_1))] \\ &= \max [\min(0,2, \mu_{\text{STOP}}(y_1)), \min(0,8, \mu_{\text{LF}}(y_1))] \end{aligned} \quad (5.10b)$$

$$\mu_{j\text{min}}(y_2) = \min(\alpha_j, \mu_j(y_2)) \quad \text{mit} \quad j = \text{"WLL", "LLL", "NULL", "LRL", "WRL"} \quad (5.11a)$$

$$\begin{aligned} \mu_{2\text{maxmin}}(y_2) &= \max [\mu_{\text{WLLmin}}(y_2), \mu_{\text{LLLmin}}(y_2), \mu_{\text{NULLmin}}(y_2), \mu_{\text{LRLmin}}(y_2), \mu_{\text{WRLmin}}(y_2)] \\ &= \max [\min(\alpha_{\text{WLL}}, \mu_{\text{WLL}}(y_2)), \min(\alpha_{\text{LLL}}, \mu_{\text{LLL}}(y_2)), \\ &\quad \min(\alpha_{\text{NULL}}, \mu_{\text{NULL}}(y_2)), \min(\alpha_{\text{LRL}}, \mu_{\text{LRL}}(y_2)), \min(\alpha_{\text{WRL}}, \mu_{\text{WRL}}(y_2))] \\ &= \max [\min(0,8, \mu_{\text{WRL}}(y_2))] \end{aligned} \quad (5.11b)$$

5.2.6 Defuzzifizierung

Bei der Defuzzifizierung werden unscharfe in scharfe Werte konvertiert, wofür verschiedene Verfahren existieren. Neben der Methode der „Maximalen Höhe“ hat sich

in der Regelungstechnik das sog. Schwerpunktsummen-Verfahren etabliert [199] [130] [131] [132].

$$y_{1S} = \frac{\int_{y_a}^{y_e} y_1 \cdot \mu_{1maxmin}(y_1) dy_1}{\int_{y_a}^{y_e} \mu_{1maxmin}(y_1) dy_1} \quad \text{oder} \quad y_{1S} = \frac{\sum_{i=1}^n y_{1Si} \cdot A_{1i}}{\sum_{i=1}^n A_{1i}} = 2,76 \quad (5.12) \quad (5.13)$$

$$y_{2S} = \frac{\int_{y_a}^{y_e} y_2 \cdot \mu_{2maxmin}(y_2) dy_2}{\int_{y_a}^{y_e} \mu_{2maxmin}(y_2) dy_2} \quad \text{oder} \quad y_{2S} = \frac{\sum_{i=1}^n y_{2Si} \cdot A_{2i}}{\sum_{i=1}^n A_{2i}} = 5 \quad (5.14) \quad (5.15)$$

Mit Hilfe der Gleichungen (5.12) bis (5.15) können die jeweiligen Schwerpunkte y_{1S} und y_{2S} der in **Abb. 5.6** gezeigten resultierenden Zugehörigkeitsfunktionen $\mu_{1maxmin}(y_1)$ und $\mu_{2maxmin}(y_2)$ ermittelt werden. Während die Gleichungen (5.12) und (5.14) die Berechnung über die gesamten resultierenden Zugehörigkeitsfunktionen durchführt, erlauben die Gleichungen (5.13) und (5.15) die jeweiligen Teilflächen und deren Schwerpunkte aufzuaddieren, wobei n deren Anzahl ist [193]. Als resultierende Schwerpunkte erhält man für $y_{1S} = 2,76$ und für $y_{2S} = 5$.

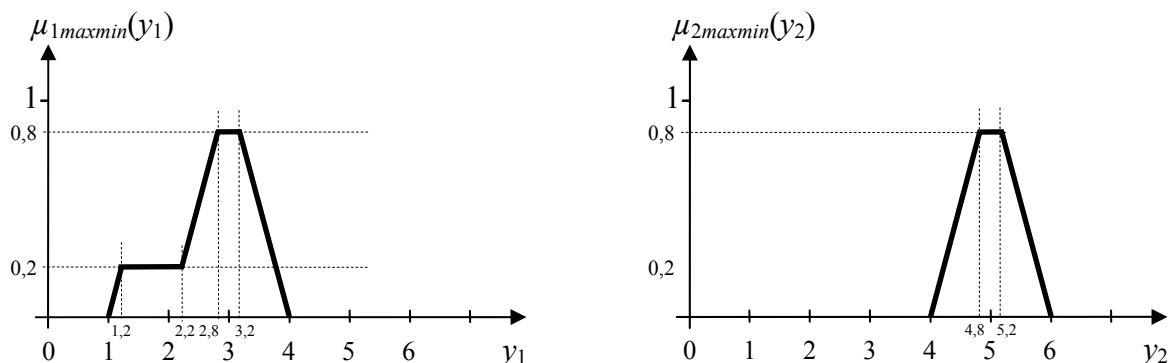


Abb. 5.6: Aggregation (Überlagerung) der Zugehörigkeitsfunktionen für die Geschwindigkeit (l.) und für den Lenkwinkel (r.).

Die hier beschriebenen Fuzzy-Logic-Regler wurden jeweils mit Matlab [51] und mit der darin enthaltenen Fuzzy-Logic-Toolbox erstellt, simuliert und als Matlab-Daten-File exportiert. Dieser File kann zusammen mit einem von der Toolbox bereitgestellten C-Code, der sog. Fuzzy-Inference-Engine [51], auf dem Zielsystem implementiert werden. Diese Implementierung stellt jeweils einen Fuzzy-Logic-Regler dar.

5.2.7 Konvertierung der Ausgangsgrößen bzw. Stellgrößen

In den **Abb. 5.4 l.**, **5.5 l.** u. **5.6 l.** wird die Geschwindigkeit y_1 sowie in den **Abb. 5.4 r.**, **5.5 r.** und **5.6 r.** wird die Lenkrichtung y_2 als Stellgröße mit $0 \leq y_1 \leq 6$ und

$0 \leq y_2 \leq 6$ ausgegeben. Abhängig von der Arbeitsweise der Motor-Control-Boards MD03 und MD22 [128] [129] müssen y_1 und y_2 nun für den Differential- und den Ackermannantrieb unterschiedlich aufbereitet werden.

Am Eingang benötigen beide Motor-Control-Boards PWM-Signale im TTL-Standard mit einem Tastverhältnis von 50% (s. Kap. 4.3.1). Diese Mittelstellung markiert die Ruhestellung der Antriebsmotoren und des Servo-Lenkantriebs, d.h. beim PWM-Tastverhältnis von $< 50\%$, $= 50\%$ oder $> 50\%$ sollen die Antriebsmotoren rückwärts, nicht oder vorwärts drehen bzw. der Servo-Lenkantrieb links, nicht oder rechts lenken.

Für den Ackermannantrieb ist ein einzelner Fahrtriebsmotor mit dem Motor-Control-Board MD03 ausreichend. Für die Lenkung des Ackermannantriebs wird ein Servo-Lenkantrieb eingesetzt, der kein zusätzliches Motor-Control-Board benötigt.

$$PWM_v(y_1) = \frac{50\%}{4} \cdot y_1 + 25\% \quad (5.16)$$

$$PWM_\phi(y_2) = \frac{50\%}{3} \cdot y_2 \quad (5.17)$$

Die oben angeführten Gleichungen (5.16) und (5.17) konvertieren die Ausgangsgrößen y_1 und y_2 in die geeigneten PWM-Tastverhältnisse für den Ackermannantrieb (s. **Abb. 5.7**). Hier werden separat die PWM-Signale jeweils für den Fahrtriebsmotor $PWM_v(y_1)$ und für den Servo-Lenkantrieb $PWM_\phi(y_2)$ erzeugt.

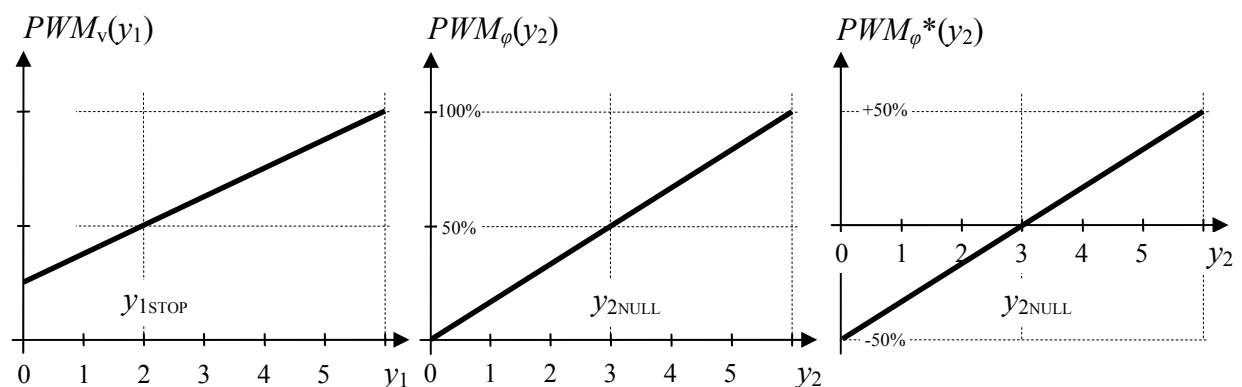


Abb. 5.7: Konvertierung der Ausgangsgrößen y_1 und y_2 für Geschwindigkeit (l.) und Lenkwinkel (M.) als PWM-Signale mit entsprechenden Tastverhältnissen $PWM_v(y_1)$ und $PWM_\phi(y_2)$ sowie die Links- / Rechtslenkung $PWM_\phi^*(y_2)$ (r.).

Als Hardware für den Differentialantrieb wurde das Motor-Control-Board MD22 gewählt, zwecks Ansteuerung des linken und des rechten Antriebsmotors. Die

Konvertierungen von y_1 und y_2 in geeignete PWM-Tastverhältnisse nehmen die Gleichungen (5.19) und (5.20) vor. Da die Lenkung beim Differentialantrieb lediglich aus der Differenz der jeweiligen Drehgeschwindigkeiten der beiden Antriebsmotoren resultiert, muss die Links- / Rechtslenkung mit Gleichung (5.18) einbezogen werden (s. **Abb. 5.7** r.).

$$PWM_{\varphi}^*(y_2) = \frac{50\%}{3} \cdot y_2 - 50\% \quad (5.18)$$

$$PWM_l(y_1, y_2) = \frac{50\%}{4} \cdot y_1 + 25\% + k \cdot PWM_{\varphi}^*(y_2) \quad (5.19)$$

$$PWM_r(y_1, y_2) = \frac{50\%}{4} \cdot y_1 + 25\% - k \cdot PWM_{\varphi}^*(y_2) \quad (5.20)$$

Das Linkslenken $0 \leq y_2 < 3$, Nichtlenken $y_2 = 3$ und Rechtslenken $3 < y_2 \leq 6$ wird in **Abb. 5.7** r. dargestellt. Gegenüber von $PWM_{\varphi}(y_2)$ gibt $PWM_{\varphi}^*(y_2)$ die Lenkrichtung nach links bzw. nach rechts mit negativem bzw. positivem Vorzeichen wieder. Mit der Konstanten k , im Normalfall $k = 1$ gesetzt, kann in den Gleichungen (5.19) und (5.20) die Stärke der Lenkrichtung den realen Gegebenheiten angepasst werden. Somit kann u.a. mit diesem Faktor der in Kap. 5.2 beschriebene Typ einer Hindernisvermeidung eingestellt werden.

Entsprechend dieser vorzeichenbehafteten Lenkrichtung wird nun beim Linkslenken das PWM-Tastverhältnis PWM_{φ}^* vom PWM_l -Signal des linken Antriebsmotors subtrahiert (5.19) und auf das PWM_r -Signal des rechten Antriebsmotors addiert (5.20), sowie umgekehrt für das Rechtslenken.

6 Selbstlokalisierung

Auch wenn ein mobiles Fahrzeug lediglich ferngesteuert wird, ist die Ermittlung des aktuellen Aufenthaltsorts mindestens einmal notwendig, und zwar dann, wenn dieses Fahrzeug nach Beendigung der Fahrt durch den Operator geborgen werden soll. Dies gilt auch für autonom agierende mobile Roboter, die nur durch Zufallsfahren (s. Kap. 5.2 und 7.2.4) eine Serviceleistung ausführen, wie z.B. Staubsauger- oder Rasenmäherroboter. Zufallsfahrende mobile Roboter können auch für Suchaufgaben eingesetzt werden, die ohne spezielle Suchstrategie Objekte wie z.B. Minen auffinden sollen. Hier ist der aktuelle Aufenthaltsort dann von Interesse, wenn das Suchobjekt „Mine“ gefunden wurde, d.h. der aktuelle Aufenthaltsort des Roboters entspricht der Position des Suchobjekts und kann in eine Karte übernommen werden.

Obige Beispiele zeigen, dass die Information über den aktuellen Aufenthaltsort des mobilen Fahrzeugs punktuell bzw. einmalig benötigt wird. Dies verhält sich bei anspruchsvolleren Aufgabenstellungen anders: Soll ein mobiles Fahrzeug entlang eines vorgegebenen Pfads fahren, so bedarf es eines kontinuierlichen Datenstroms über dessen aktuellen Aufenthaltsort. Mittels einer Bahnregelung werden die Daten des aktuellen Aufenthaltsorts (Istwert) stets mit den vorgegebenen Pfaddaten (Sollwerten) in Übereinstimmung gebracht. Weitere Beispiele sind Pfad- bzw. Bahnfahrten, Andockmanöver, Kartengenerierung, usw. bei denen stets der aktuelle Aufenthaltsort des mobilen Fahrzeugs bekannt sein muss. Diesbezüglich sei auf die entsprechende Literatur verwiesen [3] - [22].

Die Selbstlokalisierung stellt ein Verfahren dar, in dem der eigene Aufenthaltsort, auch Pose (Position und Orientierung) genannt, des mobilen Fahrzeugs innerhalb seiner Umgebung ermittelt werden. Insbesondere während der Fahrt müssen die Position und die Orientierung des mobilen Fahrzeugs kontinuierlich bzw. quasi-kontinuierlich bestimmt werden. Letzteres meint, dass die Selbstlokalisierung in bestimmten, auch von der Fahrzeuggeschwindigkeit abhängigen, Zeitabständen erfolgen muss.

6.1 Koordinatensysteme

Alle an der Fahrzeugnavigation beteiligten Systeme benötigen einen gemeinsamen Bezugspunkt bzw. ein gemeinsames Bezugskordinatensystem, das sog. Weltkoordinatensystem. Neben dem mobilen Fahrzeug selbst und seinen Sensoren, schließt dies auch die ggf. vorhandene oder noch zu generierende Karte der Fahrzeugumgebung ein.

Das mobile Fahrzeug, seine Sensoren und die relevanten Objekte der Fahrzeugumgebung, wie z.B. künstliche Land- bzw. Referenzmarken, besitzen je ein eigenes Koordinatensystem, welches jeweils mit dem Weltkoordinatensystem durch einen Translationsvektor \mathbf{t} und durch eine Rotationsmatrix \mathbf{R} verbunden ist.

Die Koordinatensysteme der Sensoren sind jedoch nur indirekt, also nur über das Fahrzeugkoordinatensystem (s. **Abb. 6.1** und **6.2**) mit dem Weltkoordinatensystem verbunden. Sensoren dienen u.a. zur Navigation des mobilen Fahrzeugs, so dass deren Koordinatensysteme auf das des Fahrzeugs bezogen bleiben. So ist beispielsweise die Position und Orientierung nur des mobilen Fahrzeugs primär von Interesse und nicht die des dafür zuständigen Sensors. In **Abb. 6.1** ist das Koordinatensystem eines mobilen Fahrzeugs mit Differentialantrieb dargestellt, dessen Ursprung zentriert auf der Achse der Antriebsräder liegt.

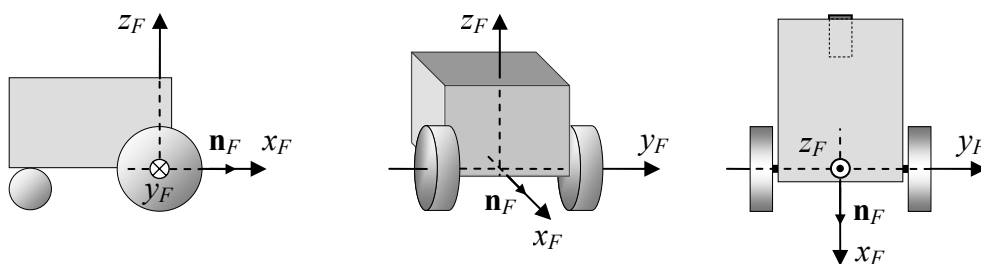


Abb. 6.1: Koordinatensystem eines Fahrzeugs.

Dementsprechend ist das Koordinatensystem der PMD-Kamera ebenfalls auf das Fahrzeugkoordinatensystem mit \mathbf{t}_K und \mathbf{R}_K bezogen, wobei der Kamerabezugspunkt im Zentrum des Objektivs liegt [194]. Damit liegen der PMD-Chip und die Optik der PMD-Kamera mit $\mathbf{t}_{chip} = (0, 0, -f)^T$ voneinander entfernt. Unter Berücksichtigung der Brennweite f , der Kalibrierungsmaßnahmen, Skalierung, zusammengefasst in intrinsische und extrinsische Entzerrung, durchlaufen die Videodaten der PMD-

Kamera weitere Transformationen, bis letztendlich die 3D-Daten der PMD-Kamera in geeigneter Form für die Bildverarbeitung vorliegen (**Abb. 6.2** u. **6.3**) [48] - [52] [194].

Aufgrund ihrer mechanischen Fixierung am mobilen Fahrzeug besitzen die Transformationsvektoren $\mathbf{t}_{Si} = (x_i, y_i, z_i)^T$ und die Rotationsmatrizen $\mathbf{R}_{Si} = \mathbf{R}_{zi} \cdot \mathbf{R}_{yi} \cdot \mathbf{R}_{xi}$ (mit $i = 1, \dots, n$ und $n = \text{Anzahl}$) der jeweiligen Sensoren quasi invariante Werte, die auf das Fahrzeugkoordinatensystem bezogen sind. \mathbf{t}_{Si} und \mathbf{R}_{Si} eines Sensors enthalten also Daten über die feste Lage des Sensorkoordinatensystems gegenüber dem Fahrzeugkoordinatensystem. Dies schließt auch bildgebende Sensoren mit \mathbf{t}_K und \mathbf{R}_K ein.

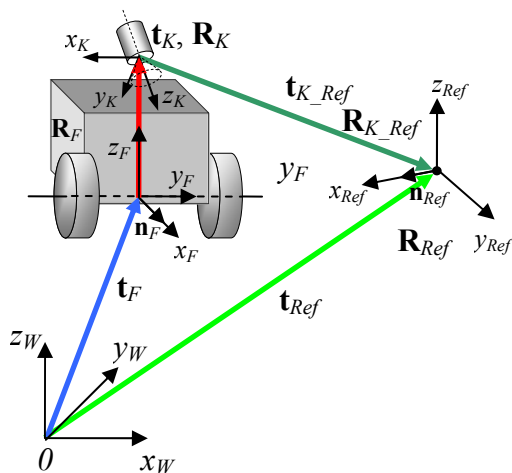


Abb. 6.2: Kamera- (K), Fahrzeug- (F) und Objektkoordinatensystem (künstliche Land- oder Referenzmarken (Ref)) sowie Weltkoordinatensystem (W).

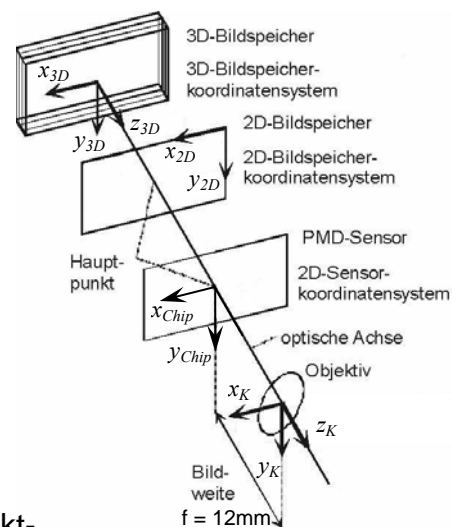


Abb. 6.3: Koordinatensysteme der PMD-Kamera [194].

Das mobile Fahrzeug kann sich sowohl in einem ebenen 2D-Areal wie auch in einem unebenen 3D-Areal bewegen. Dementsprechend gestaltet sich sein Translationsvektor \mathbf{t}_F und seine Rotationsmatrix \mathbf{R}_F des Fahrzeugkoordinatensystems gegenüber dem Weltkoordinatensystem. Bei einer Bewegung im 2D-Areal variieren lediglich die x_W - und y_W -Komponenten des Translationsvektors mit $\mathbf{t}_F = (x_W, y_W, 0)^T$ und der Rotationswinkel α_W um die z -Achse innerhalb der Rotationsmatrix mit $\mathbf{R}_F = \mathbf{R}_z(\alpha_W)$. Für Bewegungen im 3D-Areal entspricht $\mathbf{t}_F = (x_W, y_W, z_W)^T$ und $\mathbf{R}_F = \mathbf{R}_z(\alpha_W) \cdot \mathbf{R}_y(\beta_W) \cdot \mathbf{R}_x(\gamma_W)$ mit den Euler-Winkeln $\alpha_W, \beta_W, \gamma_W$ (s. Anhang 10.1.1). In Analogie zu der in der Luft- und Raumfahrt bevorzugten Bezeichnung werden diese Winkel auch Gier-Nick-Rollwinkel, englisch Yaw-Pitch-Roll-Winkel, des mobilen Fahrzeugs genannt. Bei der Selbstlokalisierung sind \mathbf{t}_F und \mathbf{R}_F genau die Werte, die die Position und Orientierung des mobilen Fahrzeugs angeben. Zur Vereinfachung soll im Weiteren der w -Index entfallen, d.h. $\mathbf{t}_F = (x, y, z)^T$ und $\mathbf{R}_F = \mathbf{R}_z(\alpha) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_x(\gamma)$.

6.2 Pose eines mobilen Fahrzeugs

Die Pose fasst die Position und die Orientierung eines mobilen Fahrzeugs zusammen und wird durch den Vektor \mathbf{p} formuliert. Die Vektorelemente stellen also die Verschiebung und die Drehung des Fahrzeug- bzw. Roboter-koordinatensystems bezogen auf das Weltkoordinatensystem dar. Für das 2D-Areal gestaltet sich die Pose als 3D-Vektor mit $\mathbf{p}_{3D} = (x, y, \alpha)^T$ und für das 3D-Areal als 6D-Vektor $\mathbf{p}_{6D} = (x, y, z, \gamma, \beta, \alpha)^T$ [83] [90] [96].

6.3 Absolute Selbstlokalisierung

Die wohl bekannteste und am meisten eingesetzte absolute Selbstlokalisierung stellt das GPS (Global Position System) [196] dar. Dieses Verfahren nutzt mindestens vier GPS-Satelliten, um lediglich die Position (keine Orientierung) des GPS-Sensors und damit des mobilen Fahrzeugs mittels Signallaufzeitmessung zu berechnen. Insgesamt kreisen 31 GPS-Satelliten in definierter Formation im nichtgeostationären Orbit. Neben den minimal erforderlichen 24 GPS-Satelliten stehen hiermit zusätzlich 7 weitere zur Verfügung, zwecks Erhöhung der Redundanz.

Wird zu den mindestens vier GPS-Satellitensignalen zusätzlich ein von einer terrestrischen Basisstation gesendetes Korrektursignal DGPS (Differential-GPS) [197] berücksichtigt, so erhält man eine Positionsgenauigkeit von 0,01 - 5 m gegenüber ca. 15 m bei GPS. Hierbei wird der Fehler, der durch die Atmosphäre oder durch Abweichungen in der Satellitenumlaufbahn verursacht wird, gleichermaßen vom GPS-Sensor am mobilen Fahrzeug und von der terrestrischen Referenzstation gemessen. Aufgrund der bekannten Position der terrestrischen Referenzstation kann der GPS-Fehler dort ermittelt und an die umliegenden DGPS-Sensoren gesendet werden.

Eine Alternative zum amerikanischen GPS bietet zukünftig das europäische Galileo-System [198]. Dennoch funktionieren Galileo, GPS oder DGPS lediglich im freien Gelände (Outdoor). Innerhalb von Gebäuden (Indoor) unterliegt das GPS-Signal hohen Signaldämpfungen und -reflektionen, so dass diese Verfahren für den Indoor-Bereich unbrauchbar sind.

Absolute Selbstlokalisierung bei einem mobilen Fahrzeug mit Hilfe eines GPS-Sensors erfolgt nun derart, dass zunächst eine markante Position, z.B. am Gebäude, im Gelände, als Ursprung eines sog. Weltkoordinatensystems mit z.B. Nord-Ausrichtung einer Koordinatenachse gewählt wird. Die Distanz zwischen der Fahrzeugposition und dem Ursprung dieses Weltkoordinatensystems lässt sich nun anhand der Längen- und Breitengrade beider Positionen ermitteln. Neuere GPS-Sensoren sind auch in der Lage, neben der Position auch die Orientierung des mobilen Fahrzeugs zu messen. Andernfalls müsste die Orientierung mit einem zusätzlichen elektronischen Kompass erfasst werden.

Für die absolute Selbstlokalisierung für den Indoor- und Outdoor-Bereich eignen sich der Einsatz von künstlichen Land- bzw. Referenzmarken mit individuellen Merkmalen, wie Textur-, Barcode-, RFID (Radio Frequency Identification)-Marken, Funkbaken, usw. die an geeigneten Stellen platziert und deren Position und Orientierung in einer Tabelle bzw. Datenbank verwaltet werden. Eine solche Tabelle bzw. Datenbank stellt bereits eine Karte in einfachster Form dar. Wird beispielsweise eine RFID-Marke durch den entsprechenden Sensor am mobilen Fahrzeug detektiert, so kann anhand der erfassten Identifikationsnummer (ID) zunächst die absolute Position und Orientierung der Referenzmarke aus der Tabelle ermittelt werden. Steht der erfassende Sensor punktgenau auf bzw. über der Referenzmarke, so ist die Fahrzeugposition identisch mit der Position der Referenzmarke. Bei Versatz, d.h. der Sensor steht nur ungenau auf bzw. über der Referenzmarke, ergeben sich Fehler bei der Bestimmung der Fahrzeugposition.

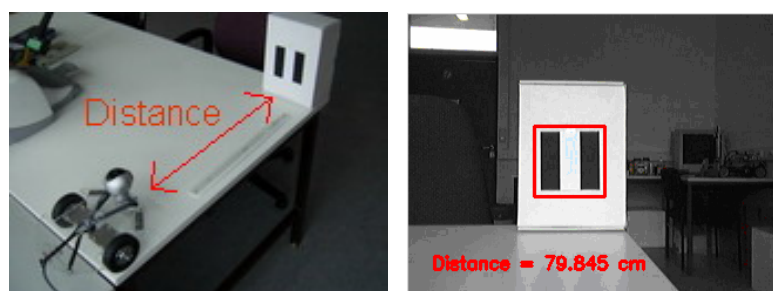


Abb. 6.4: Posebestimmung (l.) mit Hilfe einer 2D-Kamera anhand einer künstlichen Land- bzw. Referenzmarke (r.) [122] [123].

Im Gegensatz zu einer RFID-Marke kann eine texturierte Referenzmarke mit Hilfe einer einfachen 2D-Kamera aus einer größeren Entfernung erfasst werden. Hierbei wird die Textur der Referenzmarke als Videobild erfasst und mittels Bildverarbeitung

identifiziert. Anhand der erkannten Referenzmarke kann deren absolute Position und Orientierung aus der o.g. Tabelle bestimmt werden. Texturierte Referenzmarken in Verbindung mit Bildsensor und Bildverarbeitung haben auch den Vorteil, dass deren geometrische Größe und perspektivische Verzerrung innerhalb der Aufnahme detektiert werden können. Hieraus lassen sich dann die Distanz und der Blickwinkel des Bildsensors gegenüber der Referenzmarke errechnen (**Abb. 6.4** und **6.5**).



Abb. 6.5: Bestimmung der Pose aus verschiedenen Distanzen und Blickwinkeln zwischen 2D-Kamera und künstlichen Land- bzw. Referenzmarke [122] [123].

Gemäß den Arbeiten von J. Chanin [122] [123] wird im Lernmodus zunächst die Land- bzw. Referenzmarke als Suchmuster innerhalb des Bildverarbeitungssystems hinterlegt. Anschließend wird die Referenzmarke mit einer 2D-Kamera aufgenommen. Anhand des Videobildes sind Distanz und Blickwinkel der 2D-Kamera relativ zur Referenzmarke zu bestimmen. Das Bildverarbeitungssystem versucht nun mit Hilfe des Suchmusters diese Referenzmarke im Videobild zu finden. Da die geometrische Form der Referenzmarke bekannt ist, können anhand der perspektivischen Verzerrungen die Distanz und der Blickwinkel der 2D-Kamera gegenüber der Referenzmarke bestimmt werden.

Grundsätzlich vereinfacht sich die Erfassung von Distanz und Blickwinkel bei Verwendung einer PMD-Kamera erheblich, da hier sowohl PMD-Grauwert- als auch PMD-Tiefenwertbilder der Referenzmarke aufgenommen werden können. Von besonderer Bedeutung ist auch, dass man anhand des PMD-Tiefenwertbildes den Normalenvektor der Referenzmarke schnell ermittelt kann. In Verbindung mit einer

PMD-Kamera könnte eine Referenzmarke nicht nur texturiert sein, sondern sie könnte auch ein individuelles Form- bzw. Tiefenprofil besitzen. Diese Art von Referenzmarke wäre dann auch in Arealen mit Verschmutzungsgefahr einsetzbar [3] [84].

Die Position und Orientierung eines mobilen Fahrzeugs kann jetzt mit Hilfe eines Translationsvektors \mathbf{t}_F und einer Rotationsmatrix \mathbf{R}_F dargestellt werden (6.1). Hierbei ist \mathbf{n}_F der Normalenvektor des mobilen Fahrzeugs, der in dessen Fahrtrichtung und parallel zur x_F - Achse des Fahrzeugs- bzw. Roboterkoordinatensystems liegt und \mathbf{r}_F die Lage des Normalenvektors bezogen auf den Ursprung des Weltkoordinatensystems (s. **Abb. 6.1** und **6.2**).

$$\mathbf{r}_F = \mathbf{t}_F + \mathbf{R}_F \mathbf{n}_F \quad (6.1)$$

Der Translationsvektor \mathbf{t}_F und die Rotationsmatrix \mathbf{R}_F können in einer sog. homogenen Transformationsmatrix \mathbf{T}_F entsprechend (6.2) zusammengeführt werden (s. a. Anhang 10.1.3) [195] [199]. Gemäß Gleichung (6.3) muss allerdings der Vektor \mathbf{n}_F durch die Ergänzung einer 1-Komponente in einen homogenen 4x1-Vektor \mathbf{n}_{hF} umgewandelt sein. Der Vorteil dieser Art der Poseberechnung liegt in der Anwendung sämtlicher Vorschriften der Matrizenrechnung [195] [199].

$$\text{mit } \mathbf{T}_F = \left(\begin{array}{ccc|c} \mathbf{R}_F & & & \mathbf{t}_F \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \quad (6.2)$$

$$\text{und mit } \mathbf{n}_F = (1, 0, 0)^T \rightarrow \mathbf{n}_{hF} = (1, 0, 0, 1)^T \quad (6.3)$$

$$\mathbf{r}_{hF} = \mathbf{T}_F \mathbf{n}_{hF0} \quad (6.4)$$

In den Gleichungen (6.5) und (6.6) errechnet sich schließlich die resultierende Pose des mobilen Fahrzeugs aus der Position und Orientierung der Referenzmarke, der PMD-Kamera sowie der Distanz und Orientierung der PMD-Kamera gegenüber der Referenzmarke [195]. Die Transformationsmatrix \mathbf{T}_{K_Ref} in den Gleichungen (6.5) und (6.6) ist mit Hilfe der PMD-Kamera zu ermitteln.

$$\mathbf{T}_F \cdot \mathbf{T}_K \cdot \mathbf{T}_{K_Ref} = \mathbf{T}_{Ref} \quad (6.5)$$

$$\mathbf{T}_F = \mathbf{T}_K^{-1} \cdot \mathbf{T}_{K_Ref}^{-1} \cdot \mathbf{T}_{Ref} \quad (6.6)$$

Messfehler bei der absoluten Selbstlokalisierung entstehen einerseits durch fehlerhafte Installation bzw. Vermessung der jeweiligen Referenzmarke und andererseits bei deren Erfassung durch den Sensor. Das Wesentliche bei der absoluten Selbstlokalisierung ist jedoch, dass sowohl Posewerte wie auch Fehlerwerte aus vorangegangenen Posebestimmungen keinen Einfluss auf die aktuelle Berechnung haben. Eine Kumulierung des Fehlers findet hierbei nicht statt.

6.4 Relative Selbstlokalisierung

Bei der relativen Selbstlokalisierung werden lediglich Poseänderungen $\Delta \mathbf{p}_{3Dv} = (\Delta x_v, \Delta y_v, \Delta \alpha_v)^T$ und $\Delta \mathbf{p}_{6Dv} = (\Delta x_v, \Delta y_v, \Delta z_v, \Delta \gamma_v, \Delta \beta_v, \Delta \alpha_v)^T$ betrachtet. D.h. die momentane Ist-Pose des mobilen Fahrzeugs gegenüber der zuvor ermittelten Pose wird kontinuierlich bestimmt. Damit wird die Poseänderung, also die Änderung von Position und Orientierung, eines mobilen Fahrzeugs zwischen zwei zeitlich aufeinander folgenden Sensormessungen ermittelt.

Wie in Kap. 6.5 noch gezeigt wird, können für solche Sensormessungen u.a. Rad-Encoder verwendet werden, die entsprechend der Radbewegung Impulse generieren. Bei einem Differenzialantrieb kann so aus den Impulzzahlen des linken und des rechten Rad-Encoders pro definierter Zeiteinheit der zurückgelegte Weg und die Orientierungsänderung berechnet werden. Es handelt sich hierbei jeweils um ein Teilstück der Fahrzeugbewegung, welches auch den Bewegungsvektor darstellt. Durch Integration aller Poseänderungen seit Fahrbeginn erhält man die aktuelle Pose bzw. aktuelle Position und Orientierung des mobilen Fahrzeugs.

Diese Art der Posebestimmung wird auch Koppelnavigation (engl. Dead-Reckoning) genannt, mit der man darüber hinaus auch den durch das mobile Fahrzeug bereits zurückgelegten Pfad rekonstruieren kann (engl. Path Tracking) (**Abb. 6.6**) [96] [179] [187]. Nachstehende Gleichungen berechnen die Pose eines mobilen Fahrzeugs auf Basis der Koppelnavigation für Aktionen im 2D-Areal (6.7) und im 3D-Areal (6.8). Beide Gleichungen zeigen die Summation über alle Poseänderungen $\Delta \mathbf{p}_{3Dv}$ bzw. $\Delta \mathbf{p}_{6Dv}$ mit $v = 1 \dots n$ und $n =$ deren Anzahl.

$$\mathbf{p}_{3Dn} = \mathbf{p}_0 + \sum_{v=1}^n \Delta \mathbf{p}_{3Dv} = \mathbf{p}_0 + \sum_{v=1}^n (\Delta x_v, \Delta y_v, \Delta \alpha_v)^T \quad (6.7)$$

$$\mathbf{p}_{6Dn} = \mathbf{p}_0 + \sum_{v=1}^n \Delta \mathbf{p}_{6Dv} = \mathbf{p}_0 + \sum_{v=1}^n (\Delta x_v, \Delta y_v, \Delta z_v, \Delta \gamma_v, \Delta \beta_v, \Delta \alpha_v)^T \quad (6.8)$$

Die Änderung von Position und Orientierung kann auch mit einem Translationsvektor \mathbf{t}_v , einer Rotationsmatrix \mathbf{R}_v oder mit beiden zusammen als homogene Transformationsmatrix \mathbf{T}_v dargestellt werden. Damit lässt sich die aktuelle Pose \mathbf{T}_{Fn} des mobilen Fahrzeugs seit Fahrbeginn mit Daten der relativen Selbstlokalisierung entsprechend Gleichung (6.9) und (6.10) berechnen.

$$\mathbf{T}_{Fn} = \mathbf{T}_1 \cdot \mathbf{T}_2 \cdot \dots \cdot \mathbf{T}_v \cdot \dots \cdot \mathbf{T}_{n-1} \cdot \mathbf{T}_n \quad (6.9)$$

$$\mathbf{T}_{Fn} = \prod_{v=1}^n \mathbf{T}_v \quad (6.10)$$

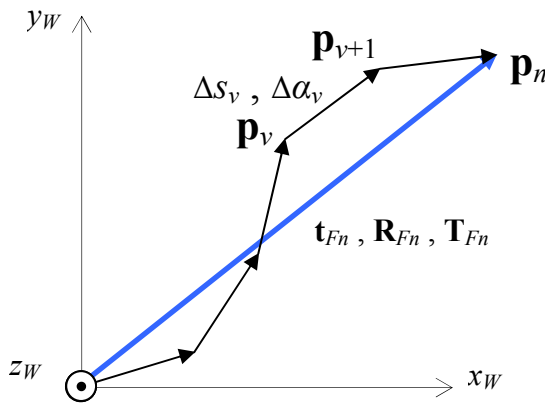


Abb. 6.6: Koppelnavigation (Dead-Reckoning).

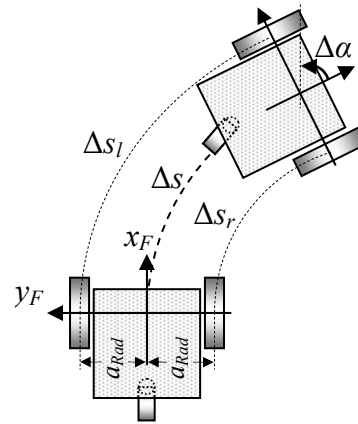


Abb. 6.7: Odometrie mit Rad-Encoder-Daten ($\Delta s_v \approx \Delta s$ und $\Delta \alpha_v \approx \Delta \alpha$).

6.5 Odometrie mit Rad-Encoder

Das einfachste und bisher am häufigsten verwendete Verfahren der Selbstlokalisierung stellt die Odometrie dar. Sie eignet sich insbesondere für Fahrten innerhalb eines ebenen 2D-Areals. Hierbei ist jedes Antriebsrad des mobilen Fahrzeugs, z.B. mit Differentialantrieb, mit einem Rad-Encoder ausgestattet. Sowohl der linke wie auch der rechte Rad-Encoder erzeugen bei Raddrehung jeweils eine bestimmte Anzahl von Impulsen pro Zeiteinheit. Mit folgenden Verhältnisgleichungen (6.11) lassen sich die Streckenstücke Δs_l und Δs_r aus den Rad-Encoder-Impulsen I_l und I_r berechnen:

$$\frac{\Delta s_l}{2\pi \cdot r_{Rad}} = \frac{I_l}{I_{max}} \quad \text{und} \quad \frac{\Delta s_r}{2\pi \cdot r_{Rad}} = \frac{I_r}{I_{max}} \quad (6.11)$$

Demnach kann die Bewegung des mobilen Fahrzeugs aus der Geometrie der Räder r_{Rad} und deren Abstand $2 \cdot a_{Rad}$ zueinander (**Abb. 6.7**) rekonstruiert werden [96]:

$$\Delta s_v \approx \Delta s = \frac{\Delta s_l - \Delta s_r}{2} \quad (6.12a)$$

$$\Delta \alpha_v \approx \Delta \alpha = \frac{\Delta s_l - \Delta s_r}{2 \cdot a_{Rad}} \quad (6.12b)$$

$$\rightarrow \quad \Delta x_v = \Delta s_v \cdot \cos\left(\frac{\Delta \alpha_v}{2} + \alpha_v\right) \quad \text{und} \quad \Delta y_v = \Delta s_v \cdot \sin\left(\frac{\Delta \alpha_v}{2} + \alpha_v\right) \quad (6.13)$$

Der in (6.12) und (6.13) eingeführte v -Index symbolisiert die fortlaufenden Poseänderungen des mobilen Fahrzeugs während der Fahrt und soll im Folgenden in diesem Sinne verwendet werden. Als Resultat erhält man eine Poseänderung $\Delta \mathbf{p}_v = (\Delta s_v, \Delta \alpha_v)^T$ in Polarkoordinatendarstellung für zunächst nur ein einzelnes Teilstück der Fahrzeugbewegung.

$$\mathbf{p}_{v+1} = \mathbf{f}(x_v, y_v, \alpha_v, \Delta s_v, \Delta \alpha_v) = \mathbf{p}_v + \Delta \mathbf{p}_v \quad (6.14)$$

$$\mathbf{p}_{v+1} = (x_v, y_v, \alpha_v)^T + (\Delta x_v, \Delta y_v, \Delta \alpha_v)^T \quad (6.15)$$

$$\mathbf{p}_{v+1} = (x_v, y_v, \alpha_v)^T + (\Delta s_v, \Delta \alpha_v)^T \quad (6.16)$$

$$\mathbf{p}_{v+1} = \mathbf{f}(x_v, y_v, \alpha_v, \Delta s_v, \Delta \alpha_v) = \begin{pmatrix} x_v + \Delta s_v \cdot \cos\left(\alpha_v + \frac{\Delta \alpha_v}{2}\right) \\ y_v + \Delta s_v \cdot \sin\left(\alpha_v + \frac{\Delta \alpha_v}{2}\right) \\ \alpha_v + \Delta \alpha_v \end{pmatrix} \quad (6.17)$$

Relative Selbstlokalisierung mit Rad-Encodern entspricht einer integrativen Methode, in der fortlaufend die inkrementellen Poseänderungen und auch sämtliche Fehler seit dem Startpunkt akkumuliert werden. Folgende Fehler fließen mit in die Berechnung ein und verfälschen die Pose des mobilen Fahrzeugs:

- Bei jeder Berechnung des Bewegungsvektors $\Delta \mathbf{p}_v = (\Delta s_v, \Delta \alpha_v)^T$ mit den Gleichungen (6.12) bis (6.17) entstehen stets Rundungsfehler.
- Des Weiteren besitzt ein Rad-Encoder eine bestimmte Auflösung, d.h. Impulsanzahl pro Umdrehung. Je höher die Auflösung, umso weniger Digitalisierungsfehler entstehen.
- Auch ist beim Antriebsrad stets mit Schlupf zu rechnen, also Rutschen bzw. Gleiten, der die Selbstlokalisierung verfälscht. Vermieden werden kann dies durch ein zusätzliches mitlaufendes Rad mit Rad-Encoder, jedoch ohne Antrieb.

- Und schließlich entstehen auch Fehler beim Überfahren von Kleinstobjekten, wie Steinchen, dünne Leitungen, kleine Absätze, usw., die auf dem Fahrweg liegen können.

6.6 Odometrie mit einer 2D-Kamera (visuelle 2D-Odometrie)

Hierzu wird eine am mobilen Fahrzeug montierte 2D-Videokamera senkrecht auf den Fahrweg gerichtet. Kontinuierlich werden zwei zeitlich aufeinander folgende 2D-Grauwertbilder **D** und **M** ausgewertet (**Abb. 6.8**). **D** und **M** stehen für „Daten“ und „Muster“ und stellen je eine $2k+1 \times 2l+1$ -Matrix mit Grauwertpixeln dar, wobei jedes einen Wertebereich von 0 - 255 besitzt. In **D** wird stets ein aktuelles 2D-Grauwertbild und in **M** wird stets ein vorheriges 2D-Grauwertbild aufgenommen. Nach B. Jähne [76] und X. Jiang et al. [203] kann daraus mit Hilfe der 2-dimensionalen Kreuzkorrelation (6.18) eine Poseänderung $\Delta \mathbf{p}_v = (\Delta s_v, \Delta \alpha_v)^T$ berechnet werden:

$$kk(i, j) = \sum_{x=-k}^k \sum_{y=-l}^l d(x, y) \cdot m(x+i, y+j) \quad (6.18)$$

Alle Ergebnisse von $kk(i, j)$ mit $i = 1, \dots, 2k+1$ und $j = 1, \dots, 2l+1$ ergeben zusammen ein Korrelationsergebnisbild, welches stets ein einzelnes Hauptmaximum und ggf. mehrere Nebenmaxima aufweist. Befindet sich dieses Hauptmaximum im Bildzentrum, d.h. $kk_{max} = kk(i_{center}, j_{center})$, so liegt keine Bewegung der 2D-Kamera bzw. des mobilen Fahrzeugs vor.

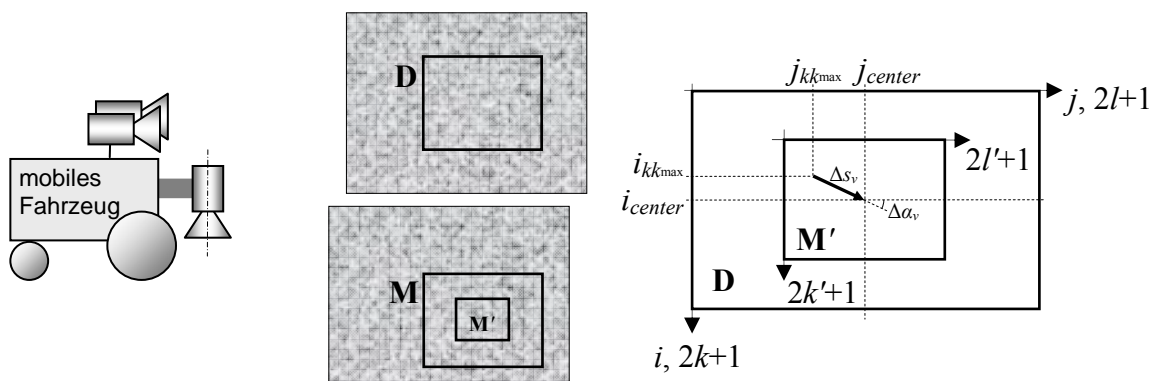


Abb. 6.8: Odometrie mit einer 2D-Kamera (l.) durch 2D-Kreuzkorrelation eines Subbildes **M'** mit einem Bild **D** einer 2D-Bildfolge der Fahrbahn (M.).

Bei einer Bewegung der 2D-Kamera bzw. des mobilen Fahrzeugs weicht das Hauptmaximum vom Bildzentrum des Korrelationsergebnisbildes ab. Je höher also

die Geschwindigkeit, desto weiter liegt das Hauptmaximum vom Zentrum entfernt. Die Distanz Δs_v und der Winkel $\Delta \alpha_v$ des Korrelationsmaximums gegenüber dem Bildzentrum entsprechen genau der gesuchten Poseänderung $\Delta \mathbf{p}_v$. Diese können mit den Gleichungen (6.19) und (6.20) berechnet werden, wobei die Konstanten c_1 und c_2 der Kalibrierung dienen.

$$\Delta s_v = \sqrt{(c_1(j_{kk \max} - j_{center}))^2 + (c_2(i_{kk \max} - i_{center}))^2} \quad (6.19)$$

$$\Delta \alpha_v = \arctan\left(\frac{c_1(j_{kk \max} - j_{center})}{c_2(i_{kk \max} - i_{center})}\right) \quad (6.20)$$

Wird nun die Geschwindigkeit des mobilen Fahrzeugs begrenzt, bleibt die Distanz zwischen Bildzentrum und Hauptmaximum kk_{\max} ebenfalls eingeschränkt, so dass der Faltungsbereich in (6.18) reduziert und damit die Berechnungszeit verringert werden kann. Um den Bewegungsvektor $\Delta \mathbf{p}_v$ mit (6.18) in Echtzeit berechnen zu können, erfolgt eine Reduktion des Korrelationsbereichs in zweierlei Hinsicht (**Abb. 6.8**):

- Das zweite 2D-Grauwertbild \mathbf{M} wird in seiner Größe reduziert, d.h. es wird ein kleineres 2D-Subbild \mathbf{M}' mit nur noch $2k'+1 \times 2l'+1$ Pixel aus dessen Bildzentrum von \mathbf{M} entnommen und mit dem ersten 2D-Grauwertbild \mathbf{D} korreliert (6.21a).
- Korrelationsoperationen, die weit entfernt vom Bildzentrum liegen, also über den o.g. Begrenzungsbereich hinaus reichen, sind im Ergebnis gleich Null. Also können in diesen Bereichen Korrelationsoperationen entfallen. Damit erfolgt die Korrelation selbst nicht mehr über den gesamten Bereich des 2D-Grauwertbildes \mathbf{D} , sondern lediglich wenige Pixel um das Bildzentrum i_{center} und j_{center} herum. Es handelt sich quasi um eine kleine Variation von wenigen Pixeln des 2D-Subbildes \mathbf{M}' gegenüber dem 2D-Grauwertbild \mathbf{D} . Diese Variationsbreite ist allerdings von der Fahrzeuggeschwindigkeit sowie von der Frame-Rate der Kamera abhängig und ist dementsprechend mit 3×3 , 5×5 , 7×7 , ... usw. Pixel anzupassen. Je niedriger die Geschwindigkeit des mobilen Fahrzeugs bzw. je höher die Frame-Rate der 2D-Kamera ist, desto kleiner kann die Variationsbreite ε gewählt und damit der Berechnungsprozess mit (6.21b) schneller durchgeführt werden.

$$kk'(i, j) = \sum_{x=i_{center}-k'}^{i_{center}+k'} \sum_{y=j_{center}-l'}^{j_{center}+l'} d(x, y) \cdot m'(x+i, y+j) \quad (6.21a)$$

$$kk'(i, j) = \sum_{x=i_{center}-\varepsilon}^{i_{center}+\varepsilon} \sum_{y=j_{center}-\varepsilon}^{j_{center}+\varepsilon} d(x, y) \cdot m'(x+i, y+j) \quad (6.21b)$$

mit $\varepsilon = 3 \vee 5 \vee 7 \vee \dots \leq k' < l'$

6.7 Kumulierung des Fehlers bei der Koppelnavigation

Die für die Posebestimmung oben angeführten Gleichungen (6.14) bis (6.17) gelten nur für fehlerlose Daten. In der Praxis erweisen sich jedoch alle Messungen und die darauf basierenden Berechnungen als fehlerbehaftet. Insbesondere bei der Koppelnavigation addieren sich Mess- und Berechnungsfehler bei jeder Poseänderung kumulativ auf. Nach längerer Pfadfahrt des mobilen Fahrzeugs verfälscht sich die so berechnete Pose zu einem unbrauchbaren Wert.

Die Berücksichtigung dieser Fehler erfordert einen probabilistischen Ansatz, wonach die Daten jedes Sensors und auch deren Weiterverarbeitung statistischen Gesetzmäßigkeiten unterliegen. Es wird davon ausgegangen, dass diese Fehler statistisch normalverteilt sind und damit einer Gaußschen-Verteilung als Fehlerdichtefunktion folgen: $x \sim N(\mu_x, \sigma_x^2)$, $y \sim N(\mu_y, \sigma_y^2)$, $\alpha \sim N(\mu_\alpha, \sigma_\alpha^2)$, $\Delta s \sim N(\mu_{\Delta s}, \sigma_{\Delta s}^2)$ und $\Delta\alpha \sim N(\mu_{\Delta\alpha}, \sigma_{\Delta\alpha}^2)$.

Eine adäquate Fehlermodellierung findet sich bei R. Siegwart et al. [96] und in [204] - [209] [237] [238], die die statistischen Abhängigkeiten der Sensordaten durch Verwendung von Kovarianzen einbeziehen. Die Pose \mathbf{p}_v des mobilen Fahrzeugs wird durch die x_v -, y_v -Koordinaten und den Orientierungswinkel α_v als untereinander statistisch abhängige Sensordaten repräsentiert. Unter Berücksichtigung der jeweiligen Fehler lässt sich $\mathbf{p}_v \sim N_3(\boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v)$ als 3-dimensionale multivariante Normalverteilung folgendermaßen formulieren:

$$g(\mathbf{p}_v) = \frac{1}{\sqrt{(2\pi)^3 \det \boldsymbol{\Sigma}_v}} e^{-\frac{1}{2}(\mathbf{p}_v - \boldsymbol{\mu}_v)^T \boldsymbol{\Sigma}_v^{-1} (\mathbf{p}_v - \boldsymbol{\mu}_v)} \quad (6.22a)$$

$$\text{mit } \mathbf{p}_v = \begin{pmatrix} x_v \\ y_v \\ \alpha_v \end{pmatrix}, \boldsymbol{\mu}_v = \begin{pmatrix} \mu_{x_v} \\ \mu_{y_v} \\ \mu_{\alpha_v} \end{pmatrix}, \text{ Kovarianzmatrix } \boldsymbol{\Sigma}_v = \begin{pmatrix} \sigma_{x_v}^2 & \sigma_{xy_v} & \sigma_{x\alpha_v} \\ \sigma_{yx_v} & \sigma_{y_v}^2 & \sigma_{y\alpha_v} \\ \sigma_{\alpha x_v} & \sigma_{\alpha y_v} & \sigma_{\alpha_v}^2 \end{pmatrix} \quad (6.22b)$$

In analoger Weise erfolgt die Darstellung der Poseänderung $\Delta \mathbf{p}_v$ als 2-dimensionale multivariante Normalverteilung $\Delta \mathbf{p}_v \sim N_2(\boldsymbol{\mu}_{\Delta v}, \boldsymbol{\Sigma}_{\Delta v})$:

$$g(\Delta \mathbf{p}_v) = \frac{1}{\sqrt{(2\pi)^2 \det \boldsymbol{\Sigma}_{\Delta v}}} e^{-\frac{1}{2}(\Delta \mathbf{p}_v - \boldsymbol{\mu}_{\Delta v})^T \boldsymbol{\Sigma}_{\Delta v}^{-1} (\Delta \mathbf{p}_v - \boldsymbol{\mu}_{\Delta v})} \quad (6.23a)$$

$$\text{mit } \Delta \mathbf{p}_v = \begin{pmatrix} \Delta s_v \\ \Delta \alpha_v \end{pmatrix}, \boldsymbol{\mu}_{\Delta v} = \begin{pmatrix} \mu_{\Delta s_v} \\ \mu_{\Delta \alpha_v} \end{pmatrix}, \boldsymbol{\Sigma}_{\Delta v} = \begin{pmatrix} \sigma_{\Delta s_v}^2 & \sigma_{\Delta s \Delta \alpha_v} \\ \sigma_{\Delta s \Delta \alpha_v} & \sigma_{\Delta \alpha_v}^2 \end{pmatrix} \quad (6.23b)$$

Diese beiden multivarianten Normalverteilungen \mathbf{p}_v und $\Delta \mathbf{p}_v$ sind jeweils mit ihren Erwartungswerten und ihren Kovarianzmatrizen vollständig beschrieben. Entsprechend der in Kap. 6.4 gezeigten Koppelnavigation werden zur Ermittlung der aktuellen Pose eines mobilen Fahrzeugs alle Poseänderungen seit Fahrbeginn, z.B. $\mathbf{p}_0 = (x_0, y_0, \alpha_0)^T = (0, 0, 0)^T$, aufaddiert. Da neben der Pose \mathbf{p}_v auch jede Poseänderung $\Delta \mathbf{p}_v = (\Delta s_v, \Delta \alpha_v)^T$ mit Fehlern behaftet ist, addieren sich diese entsprechend den Gleichungen (10.19) - (10.24) in folgender Weise ebenfalls auf:

$$g(\mathbf{p}_{v+1}) = \frac{1}{\sqrt{(2\pi)^3 \det \boldsymbol{\Sigma}_{v+1}}} e^{-\frac{1}{2}(\mathbf{p}_{v+1} - \boldsymbol{\mu}_{v+1})^T \boldsymbol{\Sigma}_{v+1}^{-1} (\mathbf{p}_{v+1} - \boldsymbol{\mu}_{v+1})} \quad (6.24)$$

$$\text{mit } \boldsymbol{\mu}_{v+1} = \boldsymbol{\mu}_v + \boldsymbol{\mu}_{\Delta v} = \begin{pmatrix} \mu_{x_v} + \mu_{\Delta s_v} \cdot \cos\left(\frac{\mu_{\Delta \alpha_v}}{2} + \mu_{\alpha_v}\right) \\ \mu_{y_v} + \mu_{\Delta s_v} \cdot \sin\left(\frac{\mu_{\Delta \alpha_v}}{2} + \mu_{\alpha_v}\right) \\ \mu_{\alpha_v} + \mu_{\Delta \alpha_v} \end{pmatrix} \quad (6.25)$$

$$\text{und } \boldsymbol{\Sigma}_{v+1} = \nabla_v \mathbf{f}(\boldsymbol{\mu}_v) \cdot \boldsymbol{\Sigma}_v \cdot \nabla_v \mathbf{f}(\boldsymbol{\mu}_v)^T + \nabla_{\Delta v} \mathbf{f}(\boldsymbol{\mu}_{\Delta v}) \cdot \boldsymbol{\Sigma}_{\Delta v} \cdot \nabla_{\Delta v} \mathbf{f}(\boldsymbol{\mu}_{\Delta v})^T \quad (6.26)$$

Gleichung (6.26) lässt sich anhand der in Kap. 10.3.2 gezeigten nichtlinearen Transformation einer multivarianten Normalverteilung herleiten, wobei $\mathbf{f}(\boldsymbol{\mu}_v) = \mathbf{f}(\mu_{x_v}, \mu_{y_v}, \mu_{\alpha_v})$ und $\mathbf{f}(\boldsymbol{\mu}_{\Delta v}) = \mathbf{f}(\mu_{\Delta s_v}, \mu_{\Delta \alpha_v})$ der in (6.17) und (10.26) dargestellten Funktion entspricht. Damit konnte u.a. in [96] [204] gezeigt werden, dass mit fortschreitender Fahrt die Unsicherheit von Position und Orientierung des mobilen

Fahrzeugs wächst. Dieser Zuwachs äußert sich durch Vergrößerung der Werte innerhalb der Kovarianzmatrix Σ_{v+1} bei Änderung von Position und Orientierung. Dieser Unsicherheitszuwachs kann jeweils durch eine Ellipse repräsentiert werden, in der lediglich eine Schnittebene der o.g. multivariaten Normalverteilung mit konstantem Wert, z.B. $g(\mathbf{p}_{v+1}) = konst. = 1$, gezeigt wird (s.a. **Abb. 6.9**).

$$g(\mathbf{p}_{v+1}) = \frac{1}{\sqrt{(2\pi)^3 \det \Sigma_{v+1}}} e^{-\frac{1}{2}(\mathbf{p}_{v+1} - \boldsymbol{\mu}_{v+1})^T \Sigma_{v+1}^{-1} (\mathbf{p}_{v+1} - \boldsymbol{\mu}_{v+1})} = konst. \quad (6.27a)$$

$$l^2 = (\mathbf{p}_{v+1} - \boldsymbol{\mu}_{v+1})^T \Sigma_{v+1}^{-1} (\mathbf{p}_{v+1} - \boldsymbol{\mu}_{v+1}) = konst. \quad (6.27b)$$

Die jeweilige Lage und Form dieser Ellipse lässt sich bestimmen durch Berechnung der Mahalanobis-Distanz (6.27b) sowie durch Bestimmung der Eigenwerte (6.28) und der Einheitsvektoren (6.29) der Kovarianzmatrix Σ_{v+1} . Die Einheitsvektoren \mathbf{e}_1 und \mathbf{e}_2 geben die Lage der Hauptachsen der Ellipse wieder und die Distanzen l_1 und l_2 (6.30) bestimmen deren Form.

$$\det(\Sigma_{v+1} - \lambda \mathbf{E}) = 0 \quad (6.28)$$

$$(\Sigma_{v+1} - \lambda \mathbf{E}) \mathbf{e} = 0 \quad (6.29)$$

$$l_1 = l \sqrt{\lambda_1} \quad \text{und} \quad l_2 = l \sqrt{\lambda_2} \quad (6.30)$$

Die Berechnung des Winkels φ_2 zwischen dem Einheitsvektor \mathbf{e}_2 und der x-Achse kann durch eine Hauptachsentransformation der Kovarianzmatrix Σ_{v+1} erfolgen [199] [213]. Demnach sind die Hauptdiagonalelemente innerhalb der $\mathbf{R}^T \Sigma_{v+1} \mathbf{R}$ -Matrix in (6.31) $a_{11} \neq 0$, $a_{22} \neq 0$, $a_{33} \neq 0$ und die übrigen Elemente $a_{12} = a_{21} = 0$, $a_{13} = a_{31} = 0$, $a_{23} = a_{32} = 0$. Der Winkel φ_2 lässt sich dann beispielsweise aus $a_{12} = 0$ innerhalb der $\mathbf{R}^T \Sigma_{v+1} \mathbf{R}$ -Matrix (6.31) ermitteln. Der Winkel φ_{Fe2} zwischen dem Fahrzeugvektor \mathbf{n}_F und dem Einheitsvektor \mathbf{e}_2 ist mit Gleichung (6.32) berechenbar [199] [213]:

$$\mathbf{R}_z^T(-\varphi_2) \cdot \Sigma_{v+1} \cdot \mathbf{R}_z(-\varphi_2) = \mathbf{R}_z^T \cdot \begin{pmatrix} \sigma_{x_{v+1}}^2 & \sigma_{xy_{v+1}} & \sigma_{x\alpha_{v+1}} \\ \sigma_{yx_{v+1}} & \sigma_{y_{v+1}}^2 & \sigma_{y\alpha_{v+1}} \\ \sigma_{\alpha x_{v+1}} & \sigma_{\alpha y_{v+1}} & \sigma_{\alpha_{v+1}}^2 \end{pmatrix} \cdot \mathbf{R}_z = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad (6.31)$$

$$\cos \varphi_{Fe2} = \frac{\mathbf{n}_F \cdot \mathbf{e}_2}{\sqrt{\mathbf{n}_F^2 \cdot \mathbf{e}_2^2}} \quad (6.32)$$

Die vorangegangenen Betrachtungen des Unsicherheitszuwachses bei Poseänderung und Koppelnavigation lassen sich grundsätzlich auf alle anderen Verfahren der relativen Selbstlokalisierung übertragen. Dies gilt auch für das im folgenden Kap. 6.8 behandelte Verfahren „Selbstlokalisierung mit der PMD-Kamera“, soll dort jedoch nicht mehr detailliert beschrieben werden.

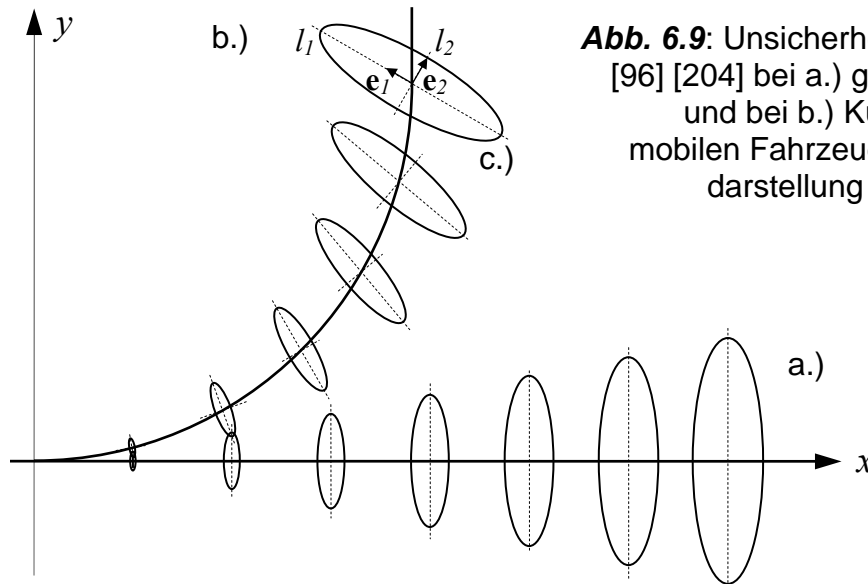


Abb. 6.9: Unsicherheitszuwachs der Pose [96] [204] bei a.) geradliniger Bewegung und bei b.) Kurvenbewegung eines mobilen Fahrzeugs, c.) Schnittebenen-darstellung der Normalverteilung.

6.8 Selbstlokalisierung mit der PMD-Kamera

Ähnlich der oben beschriebenen Odometrie mit einer 2D-Kamera könnte auch die PMD-Kamera als visuelles Odometer zur relativen Selbstlokalisierung eingesetzt werden. Die in einem mobilen Fahrzeug integrierte PMD-Kamera bildet dessen Front- bzw. Bodenbereich in den PMD-Videobildern ab. Jeweils zwei aufeinander folgende Grauwert- oder Tiefenwertbilder können entsprechend Kap. 6.6 korreliert werden, um so den Bewegungsvektor bzw. die Poseänderung zu bestimmen. Insbesondere die Auswertung von Tiefenwertbildern wurde bereits im Bereich der Medizininformatik unter Anwendung mehrdimensionaler Kreuzkorrelation realisiert, jedoch nicht für den Echtzeitbetrieb [214] [215] [216]. Für die Echtzeitfähigkeit müsste hier eine Reduzierung der Dimensionen für eine 2D-Kreuzkorrelation vollzogen werden.

Für Aktionen insbesondere in einem 3D-Areal können andere Verfahren zur Bestimmung der Poseänderung herangezogen werden, wobei sechs Freiheitsgrade zu berücksichtigen sind. Die Bewegung eines mit einer Kamera ausgestatteten

mobilen Fahrzeugs ist als entsprechender Versatz der Szene bzw. einzelner Objekte innerhalb der Videobildersequenz zu beobachten. Sicherlich könnte man Segmente und Objekte bezüglich dieses Versatzes untersuchen, jedoch zur Reduktion des Rechenaufwands soll hier lediglich der Versatz anhand natürlicher Landmarken, wie Ecken und Kanten, ermittelt werden.

Um den Versatz solcher natürlicher Landmarken innerhalb einer Videosequenz zu berechnen und daraus dann auf die Poseänderung des mobilen Fahrzeugs zu schließen, ist es von großem Vorteil, wenn die aufgenommene Szene bereits in 3D-Daten vorliegt. Abgesehen von Ultraschall- und Radar-Sensoren, können der 3D-Laserscanner wie auch die PMD-Kamera solche 3D-Daten liefern. Die PMD-Kamera stellt mit ihrer Frame-Rate von derzeit 25 Bildern/Sekunde [34] einen schnellen 3D-Bildsensor dar.

Versatzbestimmungen auf der Basis von 3D-Laserscanner-Daten wurden bereits u.a. in [27] - [31] [223] vorgestellt. Die dort verwendeten Algorithmen sollen hier nun aufgegriffen und für die PMD-Kameradaten nutzbar gemacht werden, nämlich die:

- Einheitsquaternionen-Methode [29] [199] [227] - [229] und die
- HAYAI-Matching-Methode [217] [218].

Ähnliche Verfahren zur Versatzbestimmung zwischen zwei Bildern finden sich auch in der Medizin, der Geodäsie, usw. [214] [215] [216]. Es werden dabei zwei Bilder gleicher Szene, ggf. aus multimodalen Datenquellen geliefert, in der Medizin z.B. aus der Sonographie (Ultraschall), Computertomographie (CT), Magnetresonanztomographie (MR, sog. Kernspintomographie), usw. zusammengefügt, um anhand solcher Kompositionen genauere medizinischer Diagnosen durch den Arzt durchführen zu können. Neben Translation und Rotation müssen hierbei auch Skalierung, Scherung und ggf. Verzerrungen berücksichtigt werden. Der Vorgang des Zusammenfügens wird hier Registrierung oder Matching genannt. In [214] [215] [216] wird ein Überblick über verschiedene Matching-Verfahren bezüglich Punkt-Matching (meist mit manuell generierten Kontrollpunkten), starres und elastisches Oberflächen-Matching, sowie Volumen-Matching gegeben. Die Brauchbarkeit dieser Algorithmen für die mobile Robotik ist von deren Echtzeitfähigkeit abhängig. Die meisten dieser Verfahren können zwei Bilder jedoch nur im Offline-Betrieb

registrieren bzw. „matchen“. Sie sind damit nicht echtzeitfähig und für die mobile Robotik nicht geeignet.

Demgegenüber basieren die Einheitsquaternionen- und die HAYAI-Matching-Methode darauf, dass markante Merkmale, wie Eckpunkte, innerhalb der Videobildsequenzen gefundenen und deren Versatz mittels Rotation und Translation mathematisch beschrieben werden. All dies soll in Echtzeit passieren, d.h. innerhalb derjenigen Zeit, die zwischen zwei zeitlich aufeinander folgenden Videobildern der PMD-Kamera verstreicht. Beispielsweise bei einer Frame-Rate von 10 Bilder pro Sekunde, bleibt eine Prozesszeit von nur 100 msec für Vorverarbeitung, Matching-Verfahren und Nachverarbeitung übrig.

Während die HAYAI-Matching-Methode lediglich drei Freiheitsgrade berücksichtigt, also für Fahrzeugbewegungen in einem 2D-Areal geeignet ist, verarbeitet die Einheitsquaternionen-Methode sechs Freiheitsgrade und ist damit für Fahrzeugbewegungen innerhalb einer 3D-Umgebung verwendbar. Aufgrund der reduzierten Freiheitsgrade ist die HAYAI-Matching-Methode gegenüber der Einheitsquaternionen-Methode deutlich schneller bei gleicher Rechnerleistung. Dies erlaubt eine höhere Fahrgeschwindigkeit und garantiert die Bestimmung der Poseänderung des mobilen Fahrzeugs in Echtzeit.

Doch bevor die Einheitsquaternionen- und die HAYAI-Matching-Methode näher untersucht werden, müssen die dafür notwendigen Verarbeitungsschritte vorab durchgeführt werden. Dies betrifft vor allem die Rauschunterdrückung, die Merkmalsbestimmung und insbesondere die Korrespondenzbildung, d.h. markante Merkmale im zeitlich aktuellen PMD-Videobild denen im zeitlich vorangegangenen PMD-Videobild korrekt zuzuordnen.

6.8.1 Merkmalsextraktion

Merkmalsextraktion kann grundsätzlich in Farb-, Grau- und Tiefenwertbildern durchgeführt werden. Bei Farb- und Grauwertbildern einer 2D-Kamera werden Eckpunkte aufgrund der Textur gebildet, die jedoch im hohen Maße von den Belichtungsverhältnissen einschließlich Schattenwurf beeinflusst werden. Eine kontinuierliche

Erfassung der relevanten Merkmale über mehrere Videoframes hinweg ohne solche Einflüsse ist jedoch für die relative Selbstlokalisierung unumgänglich.

Dagegen können Grauwert- und Tiefenwertbilder mit der PMD-Kamera, z.B. des Typs „PMD [vision] 3k-S“ [34], weitestgehend fremdlichtunabhängig aufgenommen werden. PMD-Videobilder sind jedoch von der Leistungs- und Leuchtcharakteristik der Beleuchtungseinheit der PMD-Kamera sowie den Reflexionseigenschaften der jeweiligen Objektoberflächen bestimmt. Abgesehen von Letzterem, können alle relevanten Merkmale mit der PMD-Kamera kontinuierlich erfasst werden, wenn stets gleich bleibende Bedingungen für die Beleuchtungseinheit herrschen.

Für die Selbstlokalisierung und ggf. später für die Kartengenerierung ist es auch notwendig, dass alle Merkmale als 3D-Information vorliegen, um deren Versatz dann als 6D-Information bestimmen zu können. Daher sollen hier für die Merkmalsextraktion ausschließlich Tiefenwertbilder der PMD-Kamera herangezogen werden.

Stehen innerhalb eines PMD-Videobildes eine zu hohe Anzahl von Merkmalen zur Verfügung, so muss eine Reduzierung dieser Anzahl auf wenige markante Merkmale erfolgen. Als Auswahlkriterium finden sich in der Literatur diverse Eigenschaften, wie Grauwerte, Eckpunkte, Kanten, Extrema, Gradienten, Segmente, usw. [76]. Die Qualität solcher markanten Merkmale soll dafür sorgen, dass diese nicht nur innerhalb eines einzelnen PMD-Videobildes extrahiert werden können, sondern die selben markanten Merkmale in den zeitlich aufeinander folgenden PMD-Videobildern stets wieder erkannt, zugeordnet (Korrespondenzbildung) und somit verfolgt werden können.

Der hier vorgestellte Ansatz konzentriert sich ausschließlich auf diejenigen Eckpunkte innerhalb eines PMD-Videobildes, die durch Kantenendungen oder -kreuzungen hervorgerufen werden. Diese markanten Merkmale sind leicht und eindeutig zu extrahieren und können schnell in den darauf folgenden PMD-Videobildern verfolgt werden. Zur Hervorhebung der Kanten wird daher jedes PMD-Videobild vorab mit dem sog. Sobel-Filter [76] bearbeitet (**Abb. 6.12.c**). Anschließend erfolgt die Extraktion der Eckpunkte mit Hilfe des sog. Moravec-Operators [225], auch Interest-Operator genannt (**Abb. 6.12.d** und **6.12.e**):

$$V_0(x,y) = \frac{1}{p(q-1)} \sum_{i=-k}^k \sum_{j=-l}^{l-1} (m(x+i,y+j) - m(x+i,y+j+1))^2 \quad (6.33)$$

$$V_{45}(x,y) = \frac{1}{(p-1)(q-1)} \sum_{i=-k}^{k-1} \sum_{j=-l}^{l-1} (m(x+i+1,y+j) - m(x+i,y+j+1))^2 \quad (6.34)$$

$$V_{90}(x,y) = \frac{1}{(p-1)q} \sum_{i=-k}^{k-1} \sum_{j=-l}^l (m(x+i,y+j) - m(x+i+1,y+j))^2 \quad (6.35)$$

$$V_{135}(x,y) = \frac{1}{(p-1)(q-1)} \sum_{i=-k}^{k-1} \sum_{j=-l}^{l-1} (m(x+i,y+j) - m(x+i+1,y+j+1))^2 \quad (6.36)$$

$$V(x,y) = \min(V_0(x,y), V_{45}(x,y), V_{90}(x,y), V_{135}(x,y)) \quad (6.37)$$

mit $p = 2k+1 = 3$ und $q = 2l+1 = 3$, sowie dem Pixel m mit den x - y -Koordinaten $(x = 1, \dots, x_{max}$ und $y = 1, \dots, y_{max})$ des Bildes \mathbf{M} . $V(x,y)$ in (6.37) stellt den jeweiligen Pixelwert des Ergebnisbildes nach der Moravec-Operation dar. Die Gleichungen (6.33) bis (6.36) zeigen, dass der Moravec-Operator stets mittlere quadratische Gradientensummen entlang der 0°-, 45°-, 90°- und 135°-Achse der Operationsmaske mit $p \times q = 3 \times 3$ Pixel bildet (s. a. **Abb. 6.10**).

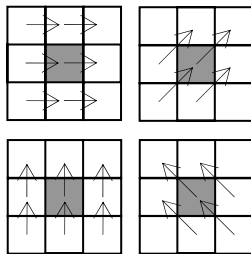


Abb. 6.10: 3x3-Moravec-Operator mit 0°-, 45°-, 90°- und 135°-Achsen.

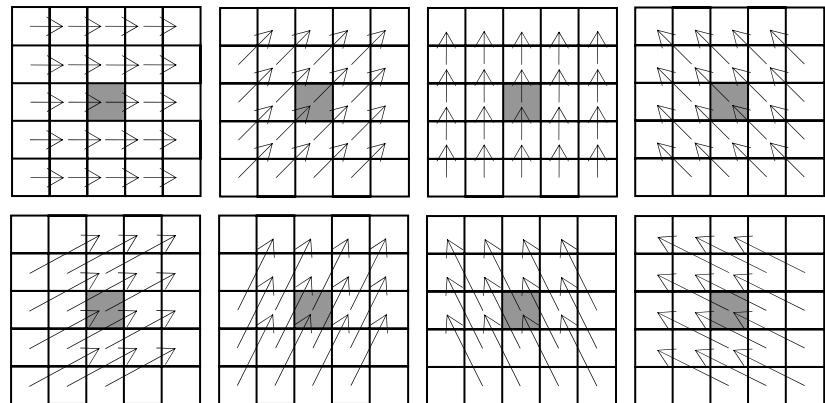


Abb. 6.11: 5x5-Moravec-Operator mit 0°-, 45°-, 90°- und 135°-Achsen (o.) sowie zusätzliche 22,5°-, 67,5°-, 112,5°- und 157,5°-Achsen (u.).

Mit lediglich vier gradientenbildenden Achsen werden Eckpunkte relativ grob detektiert. Möchte man eine genauere Detektierung der Eckpunkte, so müssten weitere Achsen, wie 22,5°, 67,5°, 112,5° und 157,5°, hinzugefügt werden. Da keine $\frac{1}{2}$ Pixel existieren, also Werte zwischen zwei Pixel, müsste man „virtuelle“ Pixel einfügen, mit jeweils dem Mittelwert der benachbarten Pixelwerte, oder man könnte stattdessen die Operationsmaske von $p \times q = 3 \times 3$ auf 5×5 Pixel vergrößern

(s. **Abb. 6.11**). Eine solche Vergrößerung der Operationsmaske würde allerdings auch zum Anwachsen des Eckpunkteareals im Ergebnisbild führen, bis hin zu einem Segment (s. **Abb. 6.12.d** und **6.12.e**).

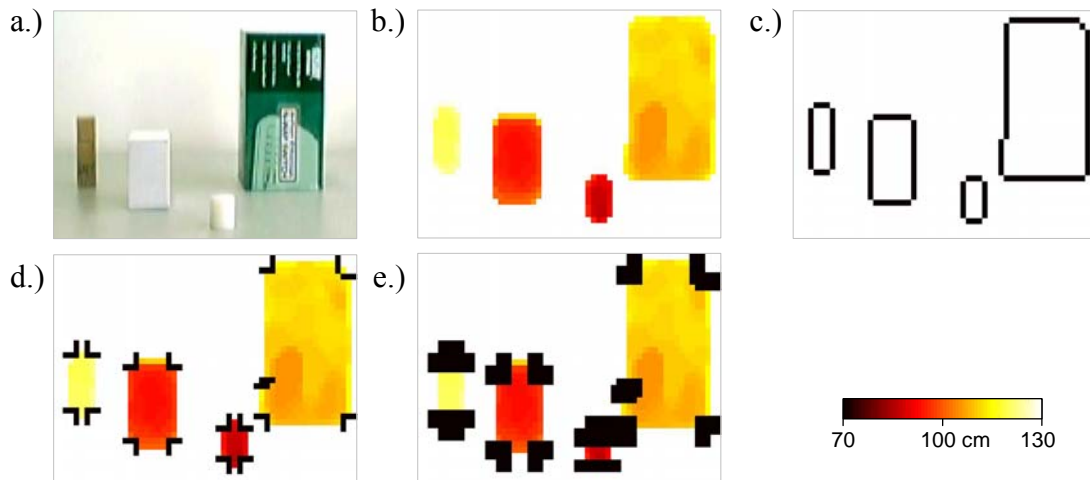


Abb. 6.12: a.) und b.) Szene mit Objekten⁸, c.) Sobel-Filterung [76], Extrahierung markanter Merkmale mittels Moravec-Operator mit d.) 3x3 und e.) 5x5 Filtermaske.

6.8.2 Korrespondenzbildung

Die Bewegung eines mit einer PMD-Kamera ausgestatteten mobilen Fahrzeugs zeigt sich im Versatz der Szene bzw. der markanten Merkmale innerhalb der zeitlich aufeinander folgenden PMD-Videobilder (s. **Abb. 6.13.a** u. **6.13.c**). Genau diesen Versatz gilt es in Form von Translation und Rotation zu bestimmen. Hier soll nun davon ausgegangen werden, dass die aufgenommene Szene selbst keine beweglichen Objekte enthält. Daraus resultiert, dass alle markanten Merkmale denselben Versatz aufweisen bzw. dieselbe Translation und Rotation besitzen.

Voraussetzung für eine Versatzbestimmung ist die Korrespondenzbildung, d.h. es muss für das jeweilige markante Merkmal \mathbf{d}_i im ersten Tiefenwertbild \mathbf{D} genau dasselbe markante Merkmal \mathbf{m}_j im zeitlich darauf folgenden zweiten Tiefenwertbild \mathbf{M} gefunden werden (**Abb. 6.13.f** u. **6.14.a**). Wenn mindestens drei solcher Paarungen ermittelt wurden, kann daraus die Poseänderung $\Delta\mathbf{p}_v$ eindeutig bestimmt werden [29]. Je mehr solcher Paarungen an dieser Berechnung beteiligt sind, desto genauer ist das Ergebnis.

⁸ Zur Objekt- / Fahrwegeseparierung wird das PMD-Videobild zunächst mit $\mathbf{R}_x(-\varphi_{PMD})$ rotiert und der Fahrweg ausgeblendet (s. Kap. 3.4). Abschließend wird das Ergebnis-PMD-Videobild mit $\mathbf{R}_x(+\varphi_{PMD})$ wieder „zurück“ rotiert.

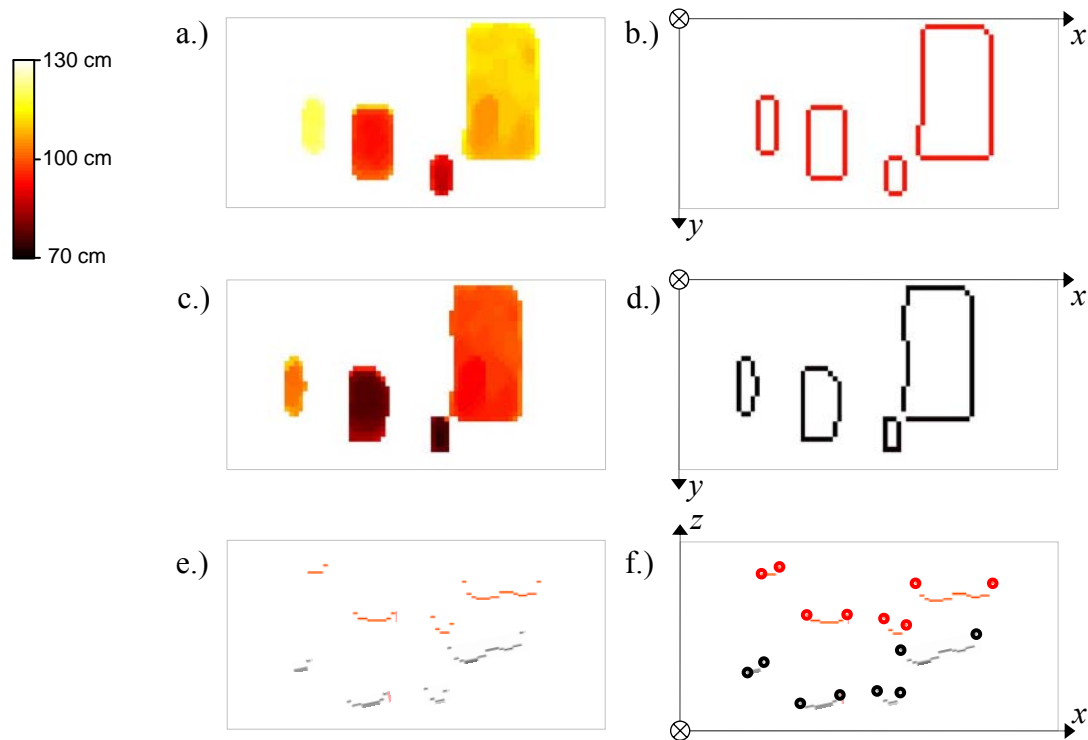


Abb. 6.13: Szeneänderung aufgrund einer Fahrzeugbewegung innerhalb einer a.) und c.) PMD-Videobildfolge⁸ und deren b.) und d.) Sobel-Filterung [76], e.) x - z -Sicht der Konturen nur einer „Höhenlinie“ und f.) x - z -Sicht der markanten Merkmale.

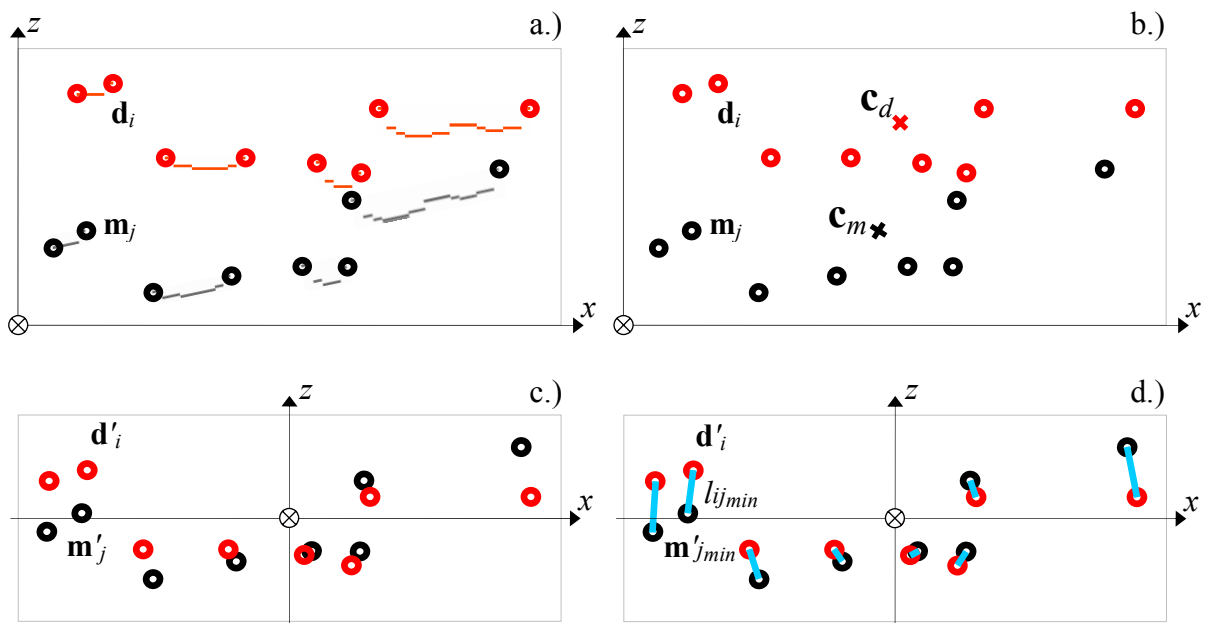


Abb. 6.14: a.) Markante Merkmale \mathbf{d}_i und \mathbf{m}_j der Konturen der in **Abb. 6.13.a** und **6.13.c** gezeigten PMD-Videobilder, b.) deren jeweiligen Bildschwerpunkte \mathbf{c}_d und \mathbf{c}_m , c.) translationsfreie Punktmenge \mathbf{d}'_i und \mathbf{m}'_j durch Überlagerung der Bildschwerpunkte \mathbf{c}_d und \mathbf{c}_m sowie d.) Korrespondenzbildung $l_i = l_{ij_{min}}$.

Zur Vermeidung von Fehlzugeordnungen bei der Korrespondenzbildung wird zunächst von den Merkmalsmengen \mathbf{d}_i und \mathbf{m}_j der beiden Tiefenwertbilder \mathbf{D} und \mathbf{M} jeweils ein Bildschwerpunkt \mathbf{c}_d und \mathbf{c}_m mit den Gleichungen (6.38) (6.39) ermittelt

(s. **Abb. 6.14.b**). Beide Tiefenwertbilder können dann mittels der Gleichungen (6.40) und (6.41) derart verschoben werden, dass sie sich in ihren jeweiligen Bildschwerpunkten überlagern (s. **Abb. 6.14.c**).

$$\mathbf{c}_d = \frac{1}{N_d} \sum_{i=1}^{N_d} \mathbf{d}_i \quad \mathbf{c}_m = \frac{1}{N_m} \sum_{j=1}^{N_m} \mathbf{m}_j \quad (6.38) \quad (6.39)$$

$$\mathbf{d}'_i = \mathbf{d}_i - \mathbf{c}_d \quad \mathbf{m}'_j = \mathbf{m}_j - \mathbf{c}_m \quad (6.40) \quad (6.41)$$

Zur Bildung von Korrespondenzen zwischen jedem Merkmal \mathbf{d}'_i und \mathbf{m}'_j beider Tiefenwertbilder \mathbf{D}' und \mathbf{M}' kommt das Verfahren der Euklidischen Distanz l_{ij} (6.42), auch Euklidische Norm genannt [199], zur Anwendung (s. **Abb. 6.14.d**) [27] - [31] [217] - [224]:

$$l_{ij} = \sqrt{\left(d'_{x_i} - m'_{x_j}\right)^2 + \left(d'_{y_i} - m'_{y_j}\right)^2 + \left(d'_{z_i} - m'_{z_j}\right)^2} \quad (6.42)$$

$$l_{ij \min} = \min_{j=1, \dots, N_m} \left(\sqrt{\left(d'_{x_i} - m'_{x_j}\right)^2 + \left(d'_{y_i} - m'_{y_j}\right)^2 + \left(d'_{z_i} - m'_{z_j}\right)^2} \right) \quad (6.43)$$

mit $i = 1, \dots, N_d$ und $j = 1, \dots, N_m$ (N_d bzw. N_m = Anzahl der markanten Merkmale in \mathbf{D}' bzw. \mathbf{M}'), sowie $i, j, N_d, N_m \in \mathbb{N}$.

Die Korrespondenzbildung zwischen \mathbf{d}'_i und \mathbf{m}'_j verläuft nun mit Hilfe der Gleichung (6.42) derart, dass zunächst vom i -ten Merkmal \mathbf{d}'_i die Distanzen l_{ij} zu allen Merkmalen \mathbf{m}'_j mit $j = 1, \dots, N_m$ berechnet werden. Daraus wird lediglich die kleinste Distanz $l_{ij \min} = \min(l_{ij})$ mit $j = 1, \dots, N_m$ ausgewählt (6.43). Die Indices i und j_{\min} der kleinsten Distanz drücken die Korrespondenz zwischen \mathbf{d}'_i und $\mathbf{m}'_{j_{\min}}$ aus. Für alle weiteren korrespondierenden Merkmalspaare \mathbf{d}'_i und \mathbf{m}'_j sind die Indices in selbiger Weise mit Gleichung (6.43) zu bestimmen.

Trotz Überlagerung der Bildschwerpunkte beider Tiefenwertbilder (6.38) - (6.41) (s. **Abb. 6.14.b** und **6.14.c**) können sich bei dieser Art der Korrespondenzbildung dennoch Fehlzuordnungen ergeben, und zwar je weiter \mathbf{d}'_i gegenüber $\mathbf{m}'_{j_{\min}}$ entfernt liegt. Zur Vermeidung der Fehlzuordnungen kann hier eine Schwellwertoperation eingeführt werden, die bei zu großer Distanz $l_{ij \min} > s$ die entsprechende Korrespondenz unterdrückt. Umgekehrt gilt aber auch, je näher \mathbf{d}'_i gegenüber $\mathbf{m}'_{j_{\min}}$ positioniert ist, desto weniger Fehlzuordnungen ergeben sich. Letzteres kann erreicht werden, wenn

die PMD-Kamera mit hoher Frame-Rate arbeitet und/oder das mobile Fahrzeug stets mit niedriger Geschwindigkeit fährt.

Bei Szeneänderungen aufgrund einer Fahrzeugbewegung können neue markante Merkmale innerhalb der PMD-Videobildfolge auftreten bzw. bekannte markante Merkmale wegfallen. Dabei erscheinen diese als Einzelmerkmale bzw. verbleiben Einzelmerkmale ohne Partnermerkmal im Folgebild bzw. im vorangegangenen PMD-Videobild. Bei der Korrespondenzbildung ergeben sich zwangsläufig Fehlzugeordnungen mit großen Distanzen l_{ij} , die jedoch mit Hilfe der Schwellwertoperation $l_{ij} > s$ aussortiert werden können.

Für den Fall, dass die Merkmalspaare \mathbf{d}'_i und $\mathbf{m}'_{j_{min}}$ relativ nahe zueinander liegen, sind die Distanzen zwischen ihnen klein und die Indices können etwa $j_{min} \approx i$ angenommen werden. Bei der Suche nach den kleinsten Distanzen $l_{ij_{min}}$ zwischen den Merkmalspaaren \mathbf{d}'_i und \mathbf{m}'_j mit Gleichung (6.43) kann jetzt der Index j statt $j = 1, \dots, N_m$ nun $j = i-3, \dots, i+3$ laufen, also nur die nächsten Nachbarmerkmale $\mathbf{m}'_{j=i-3, \dots, i, \dots, i+3}$ um $\mathbf{m}'_{j_{min}}$ herum werden hier berücksichtigt. Distanzen mit Merkmalspaaren außerhalb dieser Nachbarschaft mit $j \neq i-3, \dots, i, \dots, i+3$ berechnet, sind stets größer als die minimale Distanz $l_{ij} > l_{ij_{min}}$ und werden ohnehin mit Gleichung (6.43) ausgesondert. Somit können diese Distanzberechnungen entfallen, wodurch sich der Rechenaufwand reduziert und die Echtzeitfähigkeit eingehalten werden kann. Mit der Aussortierung der Einzelmerkmale und der Annahme $j_{min} \approx i$ kann schließlich eine Reindexierung \mathbf{d}'_i und \mathbf{m}'_i vorgenommen werden, mit $i = 1, \dots, N$, wobei $N \in \mathbb{N}$ die Anzahl der korrespondierenden markanten Merkmale darstellt.

6.8.3 Bestimmung der Transformation

Aufgrund der Fahrzeugbewegung und dem damit verbundenen Szenen-, Objekt- oder Merkmalsversatz innerhalb der zeitlich aufeinander folgenden PMD-Videobilder unterscheiden sich diese Änderungen jeweils durch einen Versatzvektor. D.h. jedes Merkmal \mathbf{d}_i im Tiefenwertbild \mathbf{D} ist gegenüber dem korrespondierenden Merkmal \mathbf{m}_i im zeitlich darauf folgenden Tiefenwertbild \mathbf{M} um die Poseänderung $\Delta \mathbf{p}_v$ versetzt. Diese Poseänderung spiegelt die Bewegung des mobilen Fahrzeugs wieder. Nach Besl und McKay [221] lässt sie sich durch einen Translationsvektor \mathbf{t} und eine

Rotationsmatrix \mathbf{R} entsprechend Gleichung (6.44) beschreiben, wobei $\mathbf{t} = (t_x, t_y, t_z)^T$ und $\mathbf{R} = \mathbf{R}_z \cdot \mathbf{R}_y \cdot \mathbf{R}_x$ (s. Gleichungen (10.1) im Anhang) gilt.

$$\mathbf{m}_i = \mathbf{R} \cdot \mathbf{d}_i + \mathbf{t} + \mathbf{e}_i \quad (6.44)$$

Fehler, die sich bei der Berechnung mit \mathbf{t} und \mathbf{R} in 6.48 ergeben, werden mit \mathbf{e}_i berücksichtigt. Oder umgekehrt, wären \mathbf{R} und \mathbf{t} bekannt, und würde man damit die Merkmalspunkte \mathbf{m}_i und \mathbf{d}_i deckungsgleich überlagern, so bliebe eine Positionsabweichung von \mathbf{e}_i übrig:

$$\mathbf{e}_i = \mathbf{m}_i - (\mathbf{R} \cdot \mathbf{d}_i + \mathbf{t}) \quad (6.45)$$

Da jedoch die Rotationsmatrix \mathbf{R} und der Translationsvektor \mathbf{t} unbekannt sind, können \mathbf{R} und \mathbf{t} folgendermaßen bestimmt werden. Unter Verwendung der „Gaußschen Methode der kleinsten Quadrate“ [199] kann Gleichung (6.45) in die Fehlerfunktion⁹ (6.46) überführt werden:

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^N |\mathbf{e}_i|^2 = \sum_{i=1}^N |\mathbf{m}_i - (\mathbf{R} \cdot \mathbf{d}_i + \mathbf{t})|^2 \quad (6.46)$$

Nach B. K. P. Horn [219], P. Besl und N. McKay [221] kann in Gleichung (6.46) der Translationsvektor \mathbf{t} vernachlässigt werden, wenn die Merkmalspunkte beider PMD-Videobilder jeweils in ihre Bildschwerpunkte (6.38) (6.39) überlagert (6.40) (6.41) werden (Kap. 6.8.2, **Abb. 6.14.b** u. **6.14.c**). Aufgrund dieser Überlagerung von \mathbf{c}_d und \mathbf{c}_m sind nun beide PMD-Videobilder „translationsfreie“, so dass in der Fehlerfunktion⁹ nur noch die Rotationsmatrix \mathbf{R} als Variable enthalten ist. Die Fehlerfunktion⁹ (6.46) kann nun durch Minimierung des Fehlers wie folgt gelöst werden:

$$E(\mathbf{R}) = \sum_{i=1}^N |\mathbf{m}'_i - \mathbf{R} \cdot \mathbf{d}'_i|^2 \rightarrow \min. \quad (6.47)$$

$$E(\mathbf{R}) = \sum_{i=1}^N |\mathbf{m}'_i|^2 - 2 \sum_{i=1}^N |\mathbf{m}'_i \cdot (\mathbf{R} \cdot \mathbf{d}'_i)| + \sum_{i=1}^N |\mathbf{R} \cdot \mathbf{d}'_i|^2 \rightarrow \min. \quad (6.48)$$

Nach dem Ausmultiplizieren des quadratischen Terms in (6.47) kann $|\mathbf{R} \cdot \mathbf{d}'_i|^2 = |\mathbf{d}'_i|^2$ ersetzt werden, da nach B. K. P. Horn [219] Distanzen bzw. Längen nach einer Rotation stets erhalten bleiben:

⁹ Um für die Fehlerfunktionen $E(\mathbf{R}, \mathbf{t})$, $E(\mathbf{R})$ oder deren Subterme in den betreffenden Gleichungen (6.46) bis (6.66) einen skalaren Wert errechnen zu können, müssen ggf. die Vektoren, Matrizen oder Quaternionen durch deren transponierte Form ersetzt werden.

$$E(\mathbf{R}) = \sum_{i=1}^N |\mathbf{m}'_i|^2 - 2 \sum_{i=1}^N |\mathbf{m}'_i \cdot (\mathbf{R} \cdot \mathbf{d}'_i)| + \sum_{i=1}^N |\mathbf{d}'_i|^2 \rightarrow \min. \quad (6.49)$$

$$\downarrow$$

$$\sum_{i=1}^N \mathbf{m}'_i \cdot (\mathbf{R} \cdot \mathbf{d}'_i) = \sum_{i=1}^N (\mathbf{m}'_i{}^T \cdot \mathbf{R}) \cdot \mathbf{d}'_i \rightarrow \max. \quad (6.50)$$

Die Fehlerfunktion⁹ (6.49) kann minimiert werden, indem der mittlere negative Term maximiert wird (6.50) [219]. Die Maximierung des negativen Terms (6.50) soll nun im Folgenden durch Verwendung der Quaternionen-Rechnung gezeigt werden.

6.8.3.a Bestimmung der Rotation mittels Einheitsquaternionen-Methode

Bei Drehungen mit Euler-Winkeln, insbesondere beim zweiten Euler-Winkel im Fall von $\alpha_y = \pi/2$, tritt eine Singularität in der Rotationsmatrix auf. Hierbei überlagern sich zwei der Drehachsen, so dass nachfolgende Drehungen nur noch mit zwei Dimensionen möglich sind. Dieser Verlust eines Freiheitsgrades bei der Rotation wird mit Gimbal Lock bezeichnet [226] [227]. Zur Vermeidung dieses Effektes soll die Quaternionen-Rechnung [29] [199] [227] - [229] verwendet werden. Hierfür müssen zunächst die markanten Merkmale \mathbf{d}'_i und \mathbf{m}'_i in Quaternionen, jeweils in Vektorform (6.51) (6.55), in Matrixform (6.53) (6.57) und deren konjugierte Form (6.52) (6.54) (6.56) (6.58), umgewandelt werden [199]:

$$\dot{\mathbf{d}}'_i = (0, d'_{x_i}, d'_{y_i}, d'_{z_i})^T, \quad \dot{\bar{\mathbf{d}}}'_i = (0, -d'_{x_i}, -d'_{y_i}, -d'_{z_i})^T \quad (6.51) (6.52)$$

$$\dot{\mathbf{D}}'_i = \begin{pmatrix} 0 & -d'_{x_i} & -d'_{y_i} & d'_{z_i} \\ d'_{x_i} & 0 & -d'_{z_i} & -d'_{y_i} \\ d'_{y_i} & d'_{z_i} & 0 & d'_{x_i} \\ -d'_{z_i} & d'_{y_i} & -d'_{x_i} & 0 \end{pmatrix}, \quad \dot{\bar{\mathbf{D}}}'_i = \begin{pmatrix} 0 & d'_{x_i} & d'_{y_i} & -d'_{z_i} \\ -d'_{x_i} & 0 & d'_{z_i} & d'_{y_i} \\ -d'_{y_i} & -d'_{z_i} & 0 & -d'_{x_i} \\ d'_{z_i} & -d'_{y_i} & d'_{x_i} & 0 \end{pmatrix} \quad (6.53) (6.54)$$

$$\dot{\mathbf{m}}'_i = (0, m'_{x_i}, m'_{y_i}, m'_{z_i})^T, \quad \dot{\bar{\mathbf{m}}}'_i = (0, -m'_{x_i}, -m'_{y_i}, -m'_{z_i})^T \quad (6.55) (6.56)$$

$$\dot{\mathbf{M}}'_i = \begin{pmatrix} 0 & -m'_{x_i} & -m'_{y_i} & m'_{z_i} \\ m'_{x_i} & 0 & -m'_{z_i} & -m'_{y_i} \\ m'_{y_i} & m'_{z_i} & 0 & m'_{x_i} \\ -m'_{z_i} & m'_{y_i} & -m'_{x_i} & 0 \end{pmatrix}, \quad \dot{\bar{\mathbf{M}}}'_i = \begin{pmatrix} 0 & m'_{x_i} & m'_{y_i} & -m'_{z_i} \\ -m'_{x_i} & 0 & m'_{z_i} & m'_{y_i} \\ -m'_{y_i} & -m'_{z_i} & 0 & -m'_{x_i} \\ m'_{z_i} & -m'_{y_i} & m'_{x_i} & 0 \end{pmatrix} \quad (6.57) (6.58)$$

Gemäß den Gleichungen (10.37) - (10.41) im Anhang kann die Rotationsmatrix \mathbf{R} in $\sum_{i=1}^N \mathbf{m}'_i \cdot (\mathbf{R} \cdot \mathbf{d}'_i)$ durch Einheitsquaternionen in folgender Weise ersetzt werden, wobei

\mathbf{d}'_i und \mathbf{m}'_i ebenfalls in Quaternionen entsprechend (6.51) (6.52) (6.55) und (6.56) umgewandelt sein müssen [29] [199] [219]:

$$\sum_{i=1}^N \mathbf{m}'_i \cdot (\mathbf{R} \mathbf{d}'_i) \rightarrow \sum_{i=1}^N \dot{\mathbf{m}}'_i \cdot (\dot{\mathbf{q}} \dot{\mathbf{d}}'_i \dot{\mathbf{q}}) \quad (6.59)$$

Anhand weiterer Umformungen kann der letzte Term in (6.59) nun derart umgewandelt werden, dass eine 4×4-Matrix \mathbf{N} entsprechend (6.60) gebildet wird [29] [199] [219].

$$\begin{aligned} \sum_{i=1}^N \dot{\mathbf{m}}'_i \cdot (\dot{\mathbf{q}} \dot{\mathbf{d}}'_i \dot{\mathbf{q}}) &= \sum_{i=1}^N (\dot{\mathbf{q}} \dot{\mathbf{m}}'_i) \cdot (\dot{\mathbf{d}}'_i \dot{\mathbf{q}}) = \sum_{i=1}^N (\dot{\mathbf{M}}'_i \dot{\mathbf{q}}) \cdot (\dot{\mathbf{D}}'_i \dot{\mathbf{q}}) = \\ &= \sum_{i=1}^N \dot{\mathbf{q}}^T \dot{\mathbf{M}}'^T_i \dot{\mathbf{D}}'_i \dot{\mathbf{q}} = \dot{\mathbf{q}}^T \left(\sum_{i=1}^N \dot{\mathbf{M}}'^T_i \dot{\mathbf{D}}'_i \right) \dot{\mathbf{q}} = \dot{\mathbf{q}}^T \left(\sum_{i=1}^N \mathbf{N}_i \right) \dot{\mathbf{q}} = \dot{\mathbf{q}}^T \mathbf{N} \dot{\mathbf{q}} \rightarrow \max. \quad (6.60) \end{aligned}$$

\mathbf{N} stellt eine Kreuzkovarianzmatrix dar und kann gemäß (6.60) mit $\mathbf{N} = \sum_{i=1}^N \dot{\mathbf{M}}'^T_i \dot{\mathbf{D}}'_i$ aufgebaut werden. Demnach können die Matrix-Elemente mit $S_{xx} = \sum_{i=1}^N m'_{xi} d'_{xi}$, $S_{xy} = \sum_{i=1}^N m'_{xi} d'_{yi}$, $S_{xz} = \sum_{i=1}^N m'_{xi} d'_{zi}$, usw. folgendermaßen erstellt werden [29] [199] [219]:

$$\mathbf{N} = \begin{pmatrix} S_{xx} + S_{yy} + S_{zz} & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & S_{xx} - S_{yy} - S_{zz} & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & -S_{xx} + S_{yy} - S_{zz} & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & -S_{xx} - S_{yy} + S_{zz} \end{pmatrix} \quad (6.61)$$

Nach wie vor ist der Term $\sum_{i=1}^N \dot{\mathbf{m}}'_i \cdot (\dot{\mathbf{q}} \dot{\mathbf{d}}'_i \dot{\mathbf{q}})$ bzw. seine Umformung $\dot{\mathbf{q}}^T \mathbf{N} \dot{\mathbf{q}}$ zu maximieren. Dies kann erreicht werden, indem zunächst die Eigenwerte λ_i von \mathbf{N} berechnet werden. Da \mathbf{N} eine 4×4-Matrix darstellt, erhält man entsprechend vier Eigenwerte $\lambda_1, \lambda_2, \lambda_3$ und λ_4 [199]. Wegen $\dot{\mathbf{q}}^T \mathbf{N} \dot{\mathbf{q}} \rightarrow \max.$ ist nun der größte Eigenwert λ_{max} auszuwählen, um damit den entsprechenden Eigenvektor $\dot{\mathbf{e}}_{max}$ mittels Gleichung (6.62) zu berechnen [29] [199] [219].

$$(\mathbf{N} - \lambda \mathbf{E}) \dot{\mathbf{e}} = \mathbf{0} \quad \text{und} \quad (\mathbf{N} - \lambda_{max} \mathbf{E}) \dot{\mathbf{e}}_{max} = \mathbf{0} \quad (6.62)$$

Der so berechnete Eigenvektor $\dot{\mathbf{e}}_{max} = (e_0, e_x, e_y, e_z)^T$ stellt die Lösung in Form einer Einheitsquaternion dar, woraus die Rotationsmatrix \mathbf{R} und die Eulerschen

Rotationswinkel entsprechend (10.42) im Anhang gewonnen werden können [29] [199] [219]:

$$\mathbf{R} = \begin{pmatrix} e_0^2 + e_x^2 - e_y^2 - e_z^2 & 2(e_x e_y - e_0 e_z) & 2(e_x e_z + e_0 e_y) \\ 2(e_x e_y + e_0 e_z) & e_0^2 - e_x^2 + e_y^2 - e_z^2 & 2(e_y e_z - e_0 e_x) \\ 2(e_x e_z - e_0 e_y) & 2(e_y e_z + e_0 e_x) & e_0^2 - e_x^2 - e_y^2 + e_z^2 \end{pmatrix} \quad (6.63)$$

$$\tan \Delta\alpha = \frac{2(e_x e_y + e_z e_0)}{e_0^2 + e_x^2 - e_y^2 - e_z^2}, \quad \sin \Delta\beta = -2(e_x e_z - e_y e_0), \quad \tan \Delta\gamma = \frac{2(e_y e_z + e_x e_0)}{e_0^2 - e_x^2 - e_y^2 + e_z^2} \quad (6.64)$$

Neben der hier beschriebenen Einheitsquaternion-Methode sei auf weitere in [29] [219] [220] [221] gezeigte Verfahren zur Lösung der Rotationsmatrix $\mathbf{R} = \mathbf{R}_z \cdot \mathbf{R}_y \cdot \mathbf{R}_x$ verwiesen, wie die Singulärwertzerlegung einer Matrix, die Orthonormal-Matrizen-Methode oder die Dualquaternion-Methode. All diese Verfahren eignen sich für 3D-Rotationen und damit für Bewegungen eines Fahrzeugs im 3D-Areal.

6.8.3.b Bestimmung der Rotation mittels HAYAI-Matching-Methode

Die HAYAI-Matching-Methode stellt eine vereinfachte Variante der Translations- und Rotationsbestimmung dar, die für Fahrzeugbewegungen in der 2D-Ebene geeignet ist. Sie basiert auf die bereits o.g. Vorverarbeitungsschritte, wie Merkmalsextraktion, Korrespondenzbildung und Überlagerung der Bildschwerpunkte. Diese Methode wurde von K. Lingemann [217] [218] als HAYAI (Highspeed And Yet Accurate Indoor/Outdoor-tracking)-Matching-Algorithmus entwickelt und konnte auf den mobilen Roboter KURT [230] implementiert werden [29] [30] [31]. Dieser Algorithmus arbeitet dort ausschließlich mit Laserscanner-Daten. In dieser vorliegenden Arbeit wurde die HAYAI-Matching-Methode bei der Verwendung von PMD-Videobilder eingesetzt.

Bewegt sich das mobile Fahrzeug lediglich in einem ebenen 2D-Areal, so reduzieren sich die Anzahl der Variablen innerhalb der Fehlerfunktion⁹ (6.47) auf zwei Translationsvariablen x und z , sowie eine Rotationsmatrix $\mathbf{R} = \mathbf{R}_y = \mathbf{R}_y(\Delta\alpha)$ mit einem Rotationswinkel $\Delta\alpha$:

$$E(\mathbf{R}_y) = \sum_{i=1}^N |\mathbf{m}'_i - \mathbf{R}_y \cdot \mathbf{d}'_i|^2 \rightarrow \min. \quad (6.65)$$

In Analogie zu den Gleichungen (6.48) bis (6.50), einschließlich der Längenerhaltung bei Rotation mit $|\mathbf{R}_y \cdot \mathbf{d}'_i|^2 = |\mathbf{d}'_i|^2$, erhält man aus (6.65) den zu maximierenden Term:

$$\sum_{i=1}^N \mathbf{m}'_i \cdot (\mathbf{R}_y \cdot \mathbf{d}'_i) \rightarrow \max. \quad (6.66)$$

Die Gleichung (6.66) besitzen damit lediglich eine einzelne Variable \mathbf{R}_y bzw. $\Delta\alpha$, sodass deren Maximum lediglich durch die erste und die zweite Ableitung nach $\Delta\alpha$ (6.67) gefunden werden kann. Als Ergebnis erhält man nach [217] [218] den Rotationswinkel $\Delta\alpha$ entsprechend Gleichung (6.68):

$$\frac{\partial E(\mathbf{R}_y)}{\partial \Delta\alpha} = \frac{\partial}{\partial \Delta\alpha} \sum_{i=1}^N \mathbf{m}'_i \cdot (\mathbf{R}_y \cdot \mathbf{d}'_i) = 0 \quad \text{und} \quad \frac{\partial^2 E(\mathbf{R}_y)}{\partial \Delta\alpha^2} < 0 \quad (6.67)$$

$$\Delta\alpha = \arctan \frac{\sum_{i=1}^N (m'_{x_i} d'_{x_i} + m'_{z_i} d'_{z_i})}{\sum_{i=1}^N (m'_{z_i} d'_{x_i} - m'_{x_i} d'_{z_i})} \quad (6.68)$$

6.8.3.c Bestimmung der Translation

Nach der Berechnung der Rotationsmatrix $\mathbf{R} = \mathbf{R}_z(\Delta\alpha) \cdot \mathbf{R}_y(\Delta\beta) \cdot \mathbf{R}_x(\Delta\gamma)$ für 3D-Fahrbewegungen (Kap. 6.8.3.a) bzw. $\mathbf{R} = \mathbf{R}_y(\Delta\alpha)$ für 2D-Fahrbewegungen (Kap. 6.8.3.b) kann schließlich die Translationsvektor \mathbf{t} bestimmt werden [29] [219] [220] [221]:

$$\mathbf{t} = \mathbf{c}_m - \mathbf{R} \cdot \mathbf{c}_d \quad (6.69)$$

Dazu werden \mathbf{c}_d und \mathbf{c}_m entsprechend (6.38) (6.39) eingesetzt und man erhält schließlich den Translationsvektor mit x -, y - und z -Komponenten entsprechend (6.70) für das 3D-Areal bzw. mit x - und z -Komponenten entsprechend (6.71) für das 2D-Areal.

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} \mathbf{c}_{m_x} \\ \mathbf{c}_{m_y} \\ \mathbf{c}_{m_z} \end{pmatrix} - \begin{pmatrix} \cos \Delta\alpha \cdot \cos \Delta\beta & \cos \Delta\alpha \cdot \sin \Delta\beta \cdot \sin \Delta\gamma - \sin \Delta\alpha \cdot \cos \Delta\gamma & \cos \Delta\alpha \cdot \sin \Delta\beta \cdot \cos \Delta\gamma + \sin \Delta\alpha \cdot \sin \Delta\gamma \\ \sin \Delta\alpha \cdot \cos \Delta\beta & \sin \Delta\alpha \cdot \sin \Delta\beta \cdot \sin \Delta\gamma + \cos \Delta\alpha \cdot \cos \Delta\gamma & \sin \Delta\alpha \cdot \sin \Delta\beta \cdot \cos \Delta\gamma - \cos \Delta\alpha \cdot \sin \Delta\gamma \\ -\sin \Delta\beta & \cos \Delta\beta \cdot \sin \Delta\gamma & \cos \Delta\beta \cdot \cos \Delta\gamma \end{pmatrix} \begin{pmatrix} \mathbf{c}_{d_x} \\ \mathbf{c}_{d_y} \\ \mathbf{c}_{d_z} \end{pmatrix} \quad (6.70)$$

$$\begin{pmatrix} \Delta x \\ \Delta z \end{pmatrix} = \begin{pmatrix} \mathbf{c}_{m_x} \\ \mathbf{c}_{m_z} \end{pmatrix} - \begin{pmatrix} \cos \Delta\alpha & \sin \Delta\alpha \\ -\sin \Delta\alpha & \cos \Delta\alpha \end{pmatrix} \cdot \begin{pmatrix} \mathbf{c}_{d_x} \\ \mathbf{c}_{d_z} \end{pmatrix} \quad (6.71)$$

6.8.4 ICP (Iterative Closest/Corresponding Points) -Methode

Nach Verwendung der Einheitsquaternionen-Methode bzw. der HAYAI-Matching-Methode zur Rotationsbestimmung \mathbf{R} und anschließender Translationsberechnung \mathbf{t} können die Ergebnisse sofort für die Ermittlung der Poseänderung $\Delta \mathbf{p}_v$

weiterverwendet werden. Allerdings besitzen \mathbf{R} und \mathbf{t} Fehlerwerte, die insbesondere bei der Koppelnavigation zu schnell anwachsenden Gesamtfehlern führen würden. Hier soll nun gezeigt werden, wie die Fehlerwerte in \mathbf{R} und \mathbf{t} mit Hilfe der ICP-Methode iterativ verkleinert werden können.

Der ICP-Algorithmus wurde in B. K. P. Horn [219], K. S. Arun et al. [220] und P. Besl et al. [221] beschrieben. Zur Verringerung von Berechnungsfehlern vereinigt der ICP-Algorithmus die o.g. Operationen in folgenden Iterationsschritten (s.a. **Abb. 6.15**):

- (a) Vorverarbeitung mit Median-Filterung (s. Kap. 2.4) und Merkmalsextraktion \mathbf{d}_i und \mathbf{m}_j (s. Kap. 6.8.1) zweier zeitlich aufeinander folgenden PMD-Videobilder \mathbf{D} und \mathbf{M} .
- (b) Bildschwerpunkt-Überlagerung \mathbf{d}'_i und \mathbf{m}'_j (s. Kap. 6.8.2).
- (c) Korrespondenzbildung \mathbf{d}'_i und \mathbf{m}'_i mit Gleichung (6.43).
- (d) Rotationsmatrix \mathbf{R} z.B. mit der Einheitsquaternionen-Methode (s. Kap. 6.8.3.a) oder mit der HAYAI-Matching-Methode (s. Kap. 6.8.3.b) bestimmen.
- (e) Translationsvektor \mathbf{t} bestimmen (s. Kap. 6.8.3.c).
- (f) Anwendung von \mathbf{R} und \mathbf{t} auf die markanten Merkmale \mathbf{d}_i des zeitlich ersten PMD-Videobildes $\mathbf{D} \rightarrow \mathbf{D}_{neu}$ bzw. $\mathbf{d}_{i_{neu}}$ und Berechnung des neuen durchschnittlichen quadratischen Fehlers E_{neu} mit $\mathbf{d}_{i_{neu}}$ und \mathbf{m}_i des zeitlich zweiten PMD-Videobildes \mathbf{M} entsprechend Gleichung (6.46).
- (g) Ermittlung des Differenzbetrags $\Delta E = |E_{neu} - E_{alt}| < s_{error}$ zwischen neuem und altem durchschnittlichen quadratischen Fehler. Fällt ΔE unter einen bestimmten Schwellwert s_{error} , so erfolgt ein Abbruch der Iteration, andernfalls wiederholt sich die Iteration ab Schritt (b) mit den markanten Merkmalen $\mathbf{d}_{i_{neu}}$ des neuen ersten PMD-Videobildes \mathbf{D}_{neu} .

Als Abbruchkriterium dieser Iteration dient der Schwellwert s_{error} , dessen Größe abhängig von der verbleibenden quadratischen Fehlerdifferenz ΔE und auch von der Iterationszeitdauer $t_{ICP} = {}^{(a)}t + \sum_{i=1}^n ({}^{(b)}t_i + {}^{(c)}t_i + \dots + {}^{(g)}t_i)$ mit n Iterationsschritten ist. D.h. aus einem

- großen bzw. kleinen Schwellwert s_{error} folgt eine
- kleine bzw. große durchschnittliche quadratische Fehlerdifferenz ΔE , und gleichzeitig eine
- lange bzw. kurze Iterationszeit t_{ICP} .

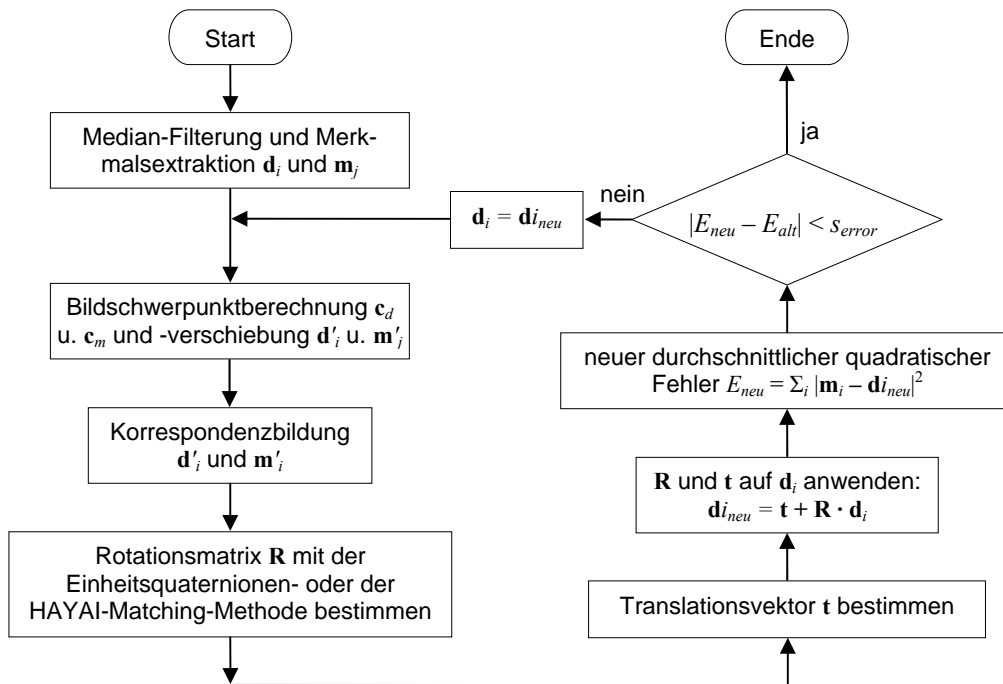


Abb. 6.15: ICP (Iterative Closest/Corresponding Points) -Methode

Die Iterationszeit t_{ICP} muss jedoch gering sein, da diese die Echtzeitfähigkeit des Selbstlokalisierungsprozesses mitbestimmt. Echtzeitfähigkeit bedeutet hier, dass die o.g. Prozessschritte (a) bis (g) und deren Wiederholungen (b) bis (g) innerhalb der Zeitdauer $t_{ICP} \leq t_{Frame}$ durchgeführt sein müssen, also in der Zeitdauer zweier aufeinander folgender PMD-Videobilder. Bei einer kleinen Iterationszeitdauer t_{ICP} hingegen bleibt eine große durchschnittliche quadratische Fehlerdifferenz ΔE übrig. Grundsätzlich ist jedoch ein möglichst kleiner Wert für ΔE anzustreben, da dieser die Genauigkeit der Pose des mobilen Fahrzeugs bei Koppelnavigation bestimmt.

6.8.5 Diskussion

Sowohl die Einheitsquaternionen- als auch die HAYAI-Matching-Methode basieren auf der Fehlerfunktion⁹ $E(\mathbf{R}) = \sum_i |\mathbf{m}'_i - \mathbf{R} \cdot \mathbf{d}'_i|^2$. Sie wurde aufgestellt, um eine Funktion von \mathbf{R} zu bekommen. Durch die Minimierung der Fehlerfunktion⁹ $E(\mathbf{R}) = \sum_i |\mathbf{m}'_i - \mathbf{R} \cdot \mathbf{d}'_i|^2 \rightarrow \min$. bzw. durch Maximierung eines darin enthaltenen negativen Subterms $\sum_i \mathbf{m}'_i \cdot (\mathbf{R} \cdot \mathbf{d}'_i) \rightarrow \max$. kann die Rotationsmatrix \mathbf{R} berechnet werden.

Bei 3D-Fahrzeugbewegungen besitzt diese Rotationsmatrix \mathbf{R} drei Euler-Winkel $\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$ als Variablen. U.a. mit Hilfe der Einheitsquaternionen-Methode kann eine

Reduzierung auf eine Variable $\Delta\varphi$ erreicht werden. In der Software-technischen Umsetzung wird dazu zunächst die 4×4 -Matrix \mathbf{N} (s. Gleichungen 6.60 u. 6.61) mit Hilfe der markanten Merkmale \mathbf{d}'_i und \mathbf{m}'_i aufgestellt, dann von \mathbf{N} der größte Eigenvektor $\hat{\mathbf{e}}_{max}$ bestimmt, um schließlich mittels dessen Komponenten e_0, e_x, e_y und e_z die drei Euler-Winkel $\Delta\alpha, \Delta\beta, \Delta\gamma$ zu berechnen (s. Gleichungen 6.64).

Demgegenüber besitzt die Rotationsmatrix \mathbf{R} bei 2D-Fahrzeugbewegungen lediglich einen einzigen Euler-Winkel $\Delta\alpha$ als Variable, sodass die Maximierung des o.g. negativen Subterms $\sum_i \mathbf{m}'_i(\mathbf{R} \cdot \mathbf{d}'_i) \rightarrow max$ durch eine einfache Kurvendiskussion und damit die Variable $\Delta\alpha$ gefunden werden kann. Dieses Verfahren wird in [217] [218] als die HAYAI-Matching-Methode beschrieben.

Im Kern berechnen beide Verfahren jeweils nur einen Rotationwinkel als Variable. Bei der Einheitsquaternionen-Methode muss dafür allerdings ein zusätzlicher Berechnungs- und damit auch Zeitaufwand in Kauf genommen werden: Prinzipiell wird hierbei eine Umwandlung der drei Euler-Winkel-Rotationen in eine Quaternionen-Rotation vorgenommen und nach der Berechnung des größten Eigenvektors erfolgt wieder die Rückumwandlung in die drei Euler-Winkel-Rotationen.

Trotz des zeitlichen Aufwands besitzt die Einheitsquaternionen-Methode den Vorteil, dass alle drei Euler-Winkel $\Delta\alpha, \Delta\beta, \Delta\gamma$ berechnet werden können. Somit kann diese Methode für 2D- und 3D-Fahrzeugbewegungen eingesetzt werden. Demgegenüber berechnet die HAYAI-Matching-Methode nur einen Euler-Winkel $\Delta\alpha$ und ist damit zunächst nur für den 2D-Fall anwendbar. Der besondere Vorteil dieser Methode liegt in ihrer Schnelligkeit bei der Berechnung des Rotationswinkels $\Delta\alpha$.

Werden nun vor der Bestimmung des Rotationswinkels $\Delta\alpha$ die Daten des Inklinationssensors berücksichtigt, dann eignet sich die HAYAI-Matching-Methode auch für 3D-Fahrzeugbewegungen: Die relativ genau gemessenen Roll- und Pitch-Winkel des Inklinationssensors (s. Kap. 4.2.3) werden vorab bei jedem PMD-Video bild „herausgerechnet“ (**Abb. 6.16**). Die beiden anhand der Erdgravitation gemessenen Winkel stellen absolute Messungen dar und brauchen bei der Koppelnavigation nicht aufaddiert zu werden. Der relativ ungenau gemessene Yaw-Winkel des Inklinationssensors ist identisch mit dem Rotations- bzw.

Orientierungswinkel α des mobilen Fahrzeugs und kann stattdessen relativ genau mittels der HAYAI-Matching-Methode entsprechend Kap. 6.8.3.b aus der PMD-Videobildfolge berechnet werden.

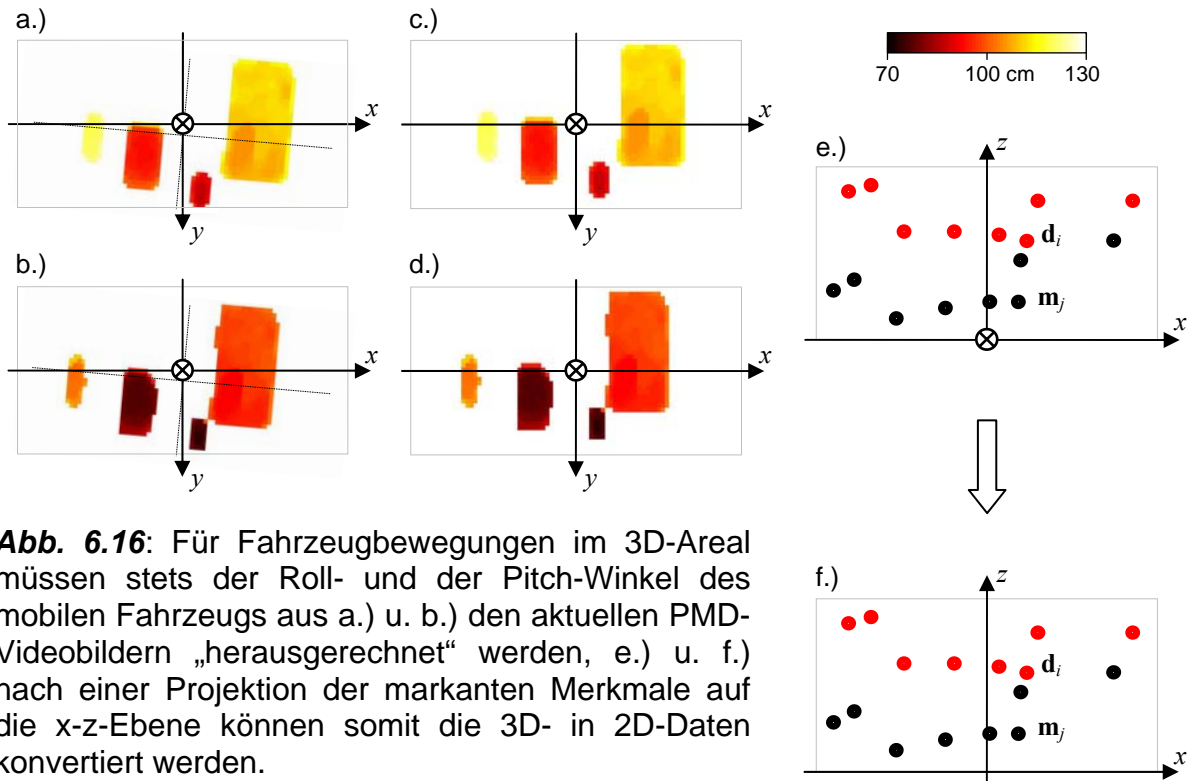


Abb. 6.16: Für Fahrzeugbewegungen im 3D-Areal müssen stets der Roll- und der Pitch-Winkel des mobilen Fahrzeugs aus a.) u. b.) den aktuellen PMD-Videobildern „herausgerechnet“ werden, e.) u. f.) nach einer Projektion der markanten Merkmale auf die x-z-Ebene können somit die 3D- in 2D-Daten konvertiert werden.

Der ICP-Algorithmus schließlich dient zur Verringerung aller Fehler der in Kap. 6.8 aufgeführten Prozessschritte zur Bestimmung der Rotationsmatrix \mathbf{R} und des Translationsvektors \mathbf{t} des mobilen Fahrzeugs.

7 Navigation und Dienstleistung

Entsprechend Kap. 1 lassen sich bereits zahlreiche Anwendungen von ferngesteuerten bis autonomen Systemen aufzählen. Der Grad der Autonomie solcher Systeme reicht von der einfachsten Fernsteuerung, also keine Autonomie, über Teilautonomie bis zur vollen Autonomie.

Prinzipiell kann man eine Fernsteuerung als Zugang eines Benutzers auf ein mobiles System sehen, wobei man von einer reinen Fernsteuerung sprechen muss, wenn keine teilautonomen bzw. autonomen Fähigkeiten vorhanden sind. Ein Mischsystem aus Fernsteuerung und autonomen Funktionen ergänzt sich zu einem teilautonomen System, indem Fernsteuerungsaktionen durch autonome Funktionen unterstützt werden. Ein vollautonomes System enthält alle nötigen autonomen Funktionen, um eine Dienstleistung eigenständig ausführen zu können. Die Fernsteuerung dient dabei zum Starten, Korrigieren, Beenden, usw. dieser Aktion durch den Operator.

Fernsteuerungsfunktionen, teilautonome und autonome Funktionen eines mobilen Systems lassen sich darüber hinaus in die Kategorien „lokale Navigation“, „globale Navigation“ und „Dienstleistung“ in hierarchischer Weise und mit fließenden Übergängen einordnen. Unter lokaler Navigation sind diejenigen Funktionen angesiedelt, die unmittelbar am mobilen Fahrzeug wirken, d.h. das Zusammenspiel von direkten Reaktionen des Systems auf Sensordaten und Kommandos. Die globale Navigation hingegen beinhaltet jene Funktion, die bei ihrer Ausführung neben den Sensordaten vor Ort auch übergeordnete externe Daten benötigt, wie z.B. Start- und Endposition eines Parcours.

Durch Kombinationen von bestehenden Funktionen entstehen neue und leistungsfähigere Funktionen, die zur Durchführung von Dienstleistungen mit unterschiedlichem Schwierigkeitsgrad eingesetzt werden können. Solche Funktionen und Dienstleistungen können ähnlich dem OSI-Referenzmodell der Kommunikationstechnik [231] hierarchisch angeordnet werden, so dass die jeweilige Funktion oder Dienstleistung stets auf untergeordneten Funktionen basiert und ihrerseits übergeordneten Funktionen dient.

Im Folgenden sollen daher Fernsteuerungsfunktionen, teilautonome und autonome Funktionen jeweils als solche beschrieben werden. In Anlehnung an die in der Fachwelt gebräuchliche Einordnung soll eine hierarchische Differenzierung in die Begriffe „lokale Navigation“, „globale Navigation“ und „Dienstleistung“ vorgenommen werden.

7.1 Lokale Navigation

7.1.1 Fahrmanöver

Bereits in Kap. 4.3.5.d wurde die Bewegungssteuerung beschrieben, die auf dem Mikrocontroller C167 implementiert wurde und via Hyper-Terminal über numerische Tasten (Pfeiltasten) am PC genutzt werden kann. Die damit möglichen Fahrmanöver bilden die einfachsten Grundfunktionen einer Fernsteuerung, wie

- Vor-, Rückwärtsfahren,
- Links-, Rechtslenken,
- Stopp,
- auf der Stelle Links- / Rechtsdrehen (beim Ackermannantrieb erfordert dies mehrere Vor- / Rückwärtsfahrten mit Links- / Rechtslenkungen).

Mit Hilfe dieser Fahrmanöver kann mit einem mobilen Fahrzeug die Dienstleistung „Transport von Gegenständen“ durchgeführt werden. Allerdings müssen bei einer reinen Fernsteuerung sowohl die Hinderniserkennung als auch die Hindernisvermeidung und alle sonstigen Navigationsaufgaben vom Operator selbst geleistet werden.

Die Kommandos dieser Fahrmanöver lassen sich über Tasten, Joystick, usw. direkt an das mobile Fahrzeug über eine drahtgebundene oder über eine drahtlose Bedieneinheit übermitteln. In Verbindung mit einer Sprachanalyseeinheit ist auch eine akustische Kommandoeingabe denkbar. Die Kommandos lassen sich aber auch via Telemetrie aus großer Distanz zwischen mobilem Fahrzeug und Operator ohne direkte visuelle Kontrolle anwenden. Die visuelle Kontrolle muss dann indirekt über eine oder mehrere Videokameras und Videoübertragung an den Operator erfolgen. In allen Fällen ist hierbei stets die visuelle Kontrolle durch den Operator maßgebend.

7.1.2 Wand folgen

Eine Wand, die links, rechts oder vor dem mobilen Fahrzeug steht, muss als solche innerhalb der PMD-Videobilder erkannt werden. Die Lenk- / Fahrregelung richtet nun das mobile Fahrzeug so aus, dass bei der Fahrbewegung die Wand stets am linken bzw. rechten äußeren Rand des PMD-Videobildes erscheint. Als Alternative kann stattdessen zusätzliche Sensorik eingesetzt werden, wie z.B. ein nach links und ein nach rechts ausgerichteter IR-Distanzsensor, jeweils um 90° zur Fahrrichtung am mobilen Fahrzeug fixiert. Hierbei reicht es aus, dass die Lenk- / Fahrregelung während der Fahrt eine konstante Distanz zur Wand ausregelt.

Die Funktion „Wand folgen“ wird für Fahrten vorzugsweise innerhalb von Gebäuden genutzt und wurde insbesondere in Staubsaugerrobotern implementiert. Auch ist es möglich, diese Funktion für die Hindernisvermeidung anzuwenden, indem das mobile Fahrzeug der Hinderniskontur folgt. Das Überschreiten des geplanten Pfads hinter dem Hindernis dient als Abbruchkriterium. Das Hindernis ist damit umfahren und das mobile Fahrzeug kann dann auf dem geplanten Pfad weiterfahren (Kap. 5.2 und **Abb. 5.1.d**).

7.1.3 Geometrische Figuren fahren

Diese Fahrzeugbewegung erfolgt ohne Selbstlokalisierung, ohne Pfadplanung und -kontrolle, sondern nur mit Hilfe einer zeitabhängigen Steuerung der Antriebsmotoren. Evtl. können solche Figuren über die Rad-Encoder-Impulse der beiden Antriebsmotoren beim Differentialantrieb per Software kontrolliert werden, wie:

- Geradenstück Vor- oder Zurückfahren,
- Kurvenstück mit definiertem Radius,
- Kreis,
- Spirale,
- Viereck,
- Schlangenlinien, usw.

Bspw. ist für große Flächenbearbeitung innerhalb des Staubsaugerroboters die spiraleförmige Fahrbewegung implementiert.

7.1.4 Hinderniserkennung und -vermeidung

Zu den fundamentalen teilautonomen Funktionen zählt die Hinderniserkennung und -vermeidung. Die Hindernisvermeidung beinhaltet Reaktionen auf zuvor von der Hinderniserkennung erfasste Hindernisse. Folgende Varianten von Reaktionen seien genannt:

- a.) permanent Stopp beim Erfassen eines Hindernisses (s. **Abb. 5.1.a**), oder
- b.) Stopp und warten, bis bewegliches Hindernis außerhalb des Erfassungsbereichs ist, danach weiterfahren,
- c.) Stopp, Richtungsänderung, weiterfahren (s. **Abb. 5.1.b**), oder
- d.) dem Hindernis ausweichen und von neuer Pose neben dem Hindernis weiterfahren (s. **Abb. 5.1.c**), oder
- e.) Hindernis umfahren und auf vorgegebenem Pfad hinter dem Hindernis weiterfahren (s. **Abb. 5.1.d**).

Die Variante a.) und b.) kommt insbesondere bei FTF als Hindernisvermeidung zum Einsatz (s. Kap. 5.2). Die Varianten c.) konnte hier mit Hilfe verschiedener Bumper (Kap. 4.2.1 und 4.2.2) und insbesondere mit Hilfe der PMD-Kamera (Kap. 3.4) und auf Basis der in Kap. 5.2 beschriebenen Hindernisvermeidung mit Fuzzy-Logic-Regelung realisiert werden.

Für die Hinderniserkennung stellt sich auch die Forderung, die unterschiedlichsten Typen von Hindernissen zu berücksichtigen (s. a. **Tab. 7.1**), wie

- stehende feste Objekte,
- stehende formveränderliche Objekte (z.B vom Wind bewegte Büsche),
- bewegliche feste Objekte,
- bewegliche formveränderliche Objekte (z.B. laufende Personen).

7.1.5 Selbstlokalisierung

Die Selbstlokalisierung, einige Methoden wurden in Kap. 6 beschrieben, erfasst die Pose des mobilen Fahrzeugs mit internen Sensoren. Selbstlokalisierung mit Hilfe externer Sensoren erfolgt beispielsweise mit einer über dem Spielfeld fixierten 2D-Kamera beim Roboterfußball [232]. Erst mit der Verfügbarkeit der Selbstlokalisierung

sind „höherwertige“ teilautonome Funktionen realisierbar, die im Folgenden als „Dienstleistungen“ zusammengefasst sind.

7.2 Globale Navigation und Dienstleistungen

In Siegwart et al. [96] und Schraft [13] werden Funktionen, wie Pfadplanung und -fahren, Kartengenerierung, usw. der „Globalen Navigation“ zugeordnet. Aufgrund der höheren Leistungsfähigkeit und Komplexität sind diese den Funktionen der „Lokalen Navigation“ hierarchisch übergeordnet.

Aus den in dieser Arbeit gewonnenen Erfahrungen kann ergänzt werden, dass die lokale und globale Navigation die eigentliche Navigation des mobilen Systems darstellt. Dagegen wird mit „Dienstleistung“ der verfolgte Zweck zum Ausdruck gebracht, wobei diese sich der lokalen und globalen Navigation sowie ggf. weiterer Funktionen und externer Daten bedient.

Teilautonome Funktionen können kreiert werden, indem z.B. die o.g. Fernsteuerungsfunktionen mit der Hinderniserkennung und -vermeidung kombiniert werden. Diese neuen Funktionen können bereits Dienstleistungen sein. Hier ist also ein fließender Übergang der Begriffe „teilautonome Funktionen“, „autonome Funktionen“ und „Dienstleistung“ feststellbar. (Teil)autonome Funktionen können auch als Dienstleistungen einfachster Art interpretiert werden.

7.2.1 Teilautonomie

Bei der Durchführung einer Dienstleistung in einer entfernten Umgebung mit Hilfe der Telepräsenz darf der Operator nicht durch Detailarbeiten zu stark belastet sein, d.h. hohe kognitive Lasten, bedingt z.B. durch Navigationsaufgaben, sind zu vermeiden. Deshalb sind sowohl für das mobile Fahrzeug wie auch für weitere Aktuatoren, wie z.B. Roboterarm mit Greifer, entsprechende teilautonome Funktionen einzurichten. Dem Operator soll dabei trotzdem ein hohes Maß an direkter Interaktion und Kontrolle erlaubt sein. Je nach aktueller Situation des mobilen Fahrzeugs soll der Operator zwischen unterschiedlichen Autonomiegraden wählen können [77] [233] - [236]. Dies gilt nicht nur für die Fernsteuerung via Telepräsenz, sondern auch für die

manuelle Steuerung (s. Kap. 7.1.1), die über Tasten, Joystick, usw. direkt am mobilen Fahrzeug ausgeübt wird.

Teilautonomie in einfachster Form bedeutet also, dass eine Funktion mit einem Start-Kommando (s. Kap. 4.3.5.d) initialisiert und die Ausführung so lange fortgesetzt wird, bis eine Endbedingung erfüllt und/oder ein Endbefehl eingegangen ist. Beispielsweise bewirkt der Befehl „30° Linksfahren“ einen entsprechenden Linkseinschlag der Lenkräder bei gleichzeitigem Vorwärtsfahren der Antriebsräder (Ackermannantrieb). Wird die gewünschte Orientierung des mobilen Fahrzeugs erreicht, erfolgt nur eine Rückstellung der Lenkräder. Die Antriebsräder bleiben erst beim Empfang des Befehls „Stopp“ stehen.

Eine Steigerung der Teilautonomie ist notwendig, wenn während der Ausführung solcher teilautonomer Funktionen ein Hindernis im Erfassungsbereich der Sensoren des mobilen Fahrzeugs auftritt. In diesem Fall wird die eigentliche teilautonome Funktion unterbrochen und stattdessen die Hindernisvermeidung aktiviert. Erst nach dem Ausweichen bzw. der Umfahrung des Hindernisses (s. Kap. 5.2, **Abb. 5.1**) wird die ursprüngliche teilautonome Funktion wieder fortgesetzt. Im Folgenden ist jede teilautonome Funktion in Kombination mit der Hindernisvermeidung zu sehen. Eine weitere Steigerung der Teilautonomie bei einfachen teilautonomen Funktionen mündet letztlich in einer gewünschten Dienstleistung, die nun diskutiert werden soll.

7.2.2 Treppen hinauf- und herabfahren

Mobile Fahrzeuge mit Ketten- bzw. Raupenantrieb (s. **Abb. 1.3**) oder Rocker-Bogie-Fahrertrieb (s. **Abb. 1.4.b** und **1.4.c**) entsprechen einem Differentialantrieb (Kap. 4.1) und können bei geeigneter Bauform auch Treppenstufen hinauf- und herabfahren. Nachteilig beim Rocker-Bogie-Fahrertrieb und insbesondere beim Kettenantrieb ist, dass diese die Stufenvorderkanten der Treppe mechanisch belasten und damit beschädigen können. Eine Alternative könnte nach **Abb. 7.1** ein Fahrwerktrieb mit vier sog. Sternradantrieben bieten. Beim Hinauf- oder Herabfahren einer Treppe erfolgen hier typischerweise nur Berührungen auf den Aufritten und an den Setzstufen der Treppenstufen.

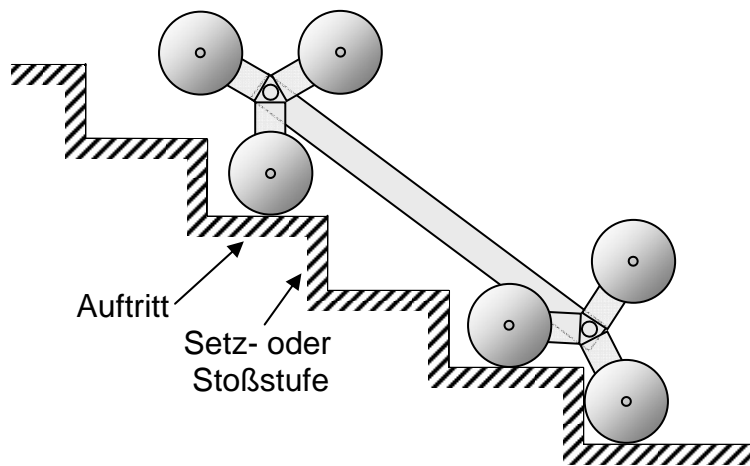


Abb.: 7.1: Schematische Darstellung eines mobilen Fahrzeugs mit Sternradantrieb.

7.2.3 Pose einnehmen

Mit Hilfe der Selbstlokalisierung können aktuelle Posedaten des mobilen Fahrzeugs stets als Istwerte ermittelt werden. Vorgaben von Orientierungen und Wegpunkten als Sollwerte durch den Operator, mit Nutzung eines geeigneten Regelungsalgorithmus, wie z.B. PID-Regler, ermöglichen das Anfahren neuer Positionen und Orientierungen mit dem mobilen Fahrzeug. Solche Positionen und Orientierungen können z.B. mit der in **Abb. 4.13** dargestellten Benutzerschnittstelle editiert und an das mobile Fahrzeug gesendet werden. Folgende teilautonome Funktionen lassen sich damit realisieren:

- **Orientierung einnehmen:** Das mobile Fahrzeug nimmt die vom Operator vorgegebene neue Orientierung ein. Während beim Differentialantrieb lediglich eine triviale Links- / Rechtsdrehung genügt, müssen beim Ackermannantrieb dafür mehrere Rangiermanöver vollzogen werden.
- **Pose anfahren:** Das mobile Fahrzeug fährt zu der vom Operator vorgegebenen einzelnen Position und nimmt dabei die gewünschte Orientierung ein.
- **Pfad- bzw. Bahnfahrt ohne Karte:** Das mobile Fahrzeug fährt nacheinander die vom Operator vorgegebenen Positionen an. Diese Positionen stellen Wegpunkte dar und bilden zusammen einen Pfad bzw. eine Bahn (s. Kap. 7.2.6).
- **UnDo-Funktion:** Werden die letzten vollzogenen Aktionen und Manöver des mobilen Fahrzeugs z.B. die letzten 10 angefahrenen Positionen stets als Historie gespeichert, können diese dem mobilen Fahrzeug nach dem FILO (First In Last Out)-Prinzip nochmals zur Aufgabe gegeben werden. Somit erhält man eine brauchbare sog. UnDo-Funktion, die die letzte(n) Aktion(en) „rückgängig“ machen.

- **Folgen:** Ohne physischen Kontakt soll das mobile Fahrzeug stets dem Operator nachfahren, indem es z.B. eine IR-Quelle, eine Referenzmarke, einen Bluetooth- oder einen WLAN-Sender einer Fernbedienungseinheit ortet und kontinuierlich folgt. Ab einer bestimmten Distanz gegenüber dem Operator muss das mobile Fahrzeug anhalten.
- **Vorausfahren:** Ähnlich der teilautonomen Funktion „Folgen“ soll hier das mobile Fahrzeug im definierten Abstand vor dem Operator wie eine Schubkarre vorher fahren, allerdings ohne physischen Kontakt.
- **Begleiten:** Ebenfalls ähnlich der teilautonomen Funktion „Folgen“ soll hier das mobile Fahrzeug links oder rechts neben dem Operator fahren. Auch hier soll kein physischer Kontakt bestehen.

7.2.4 Zufallsfahren

Mit Hilfe der teilautonomen Funktion „Stopp, Richtungsänderung, weiterfahren“ und der Fernsteuerungsfunktion „Vorwärtsfahren“ kann die Dienstleistung „Zufallsfahren“ auch ohne Selbstlokalisierung zusammengestellt werden. Beispielsweise kann damit die Suche nach Minen in einem Minenfeld mit einem mobilen System per Zufallsfahrt erfolgen. Beim Detektieren eines solchen Objekts verharret das mobile Fahrzeug vor dem Objekt und gibt damit dessen Position an.

Die Fernsteuerungsfunktion „Vorwärtsfahren“ initialisiert hierbei den Prozess und bleibt fortwährend aktiv. Beim Auftreten eines Hindernisses wird sie jedoch ausgesetzt und stattdessen wird die teilautonome Funktion „Stopp, Richtungsänderung, weiterfahren“ ausgeführt. Beim Detektieren des Suchobjekts erfolgt ein Fahrstopp.

7.2.5 Exploration und Kartengenerierung

Werden bei der Fahrt durch ein unbekanntes Areal markante Merkmale erfasst und werden diese gespeichert, spricht man hierbei von Exploration. Die Speicherung dieser Merkmale bedeutet jeweils, deren Pose und Eigenschaften in Form einer Umgebungskarte zusammen zu stellen. Dieser hier nicht weiter beschriebene

Prozess wird in der Fachwelt auch SLAM (Simultaneous Localisation and Map Building) genannt [17] [237] [238].

Neben der Selbstlokalisierung des mobilen Fahrzeugs sowie der Objektidentifizierung benötigt man dafür auch die Objektlokalisierung. Entsprechend **Abb. 6.2** und den Gleichungen (6.7) (6.8) wird die relative Pose $(\mathbf{t}_{F_Obj}, \boldsymbol{\varphi}_{F_Obj})^T$ des Objekts gegenüber dem mobilen Fahrzeug mit Hilfe z.B. bildgebender Sensoren bestimmt. Diese aufaddiert mit der Pose $(\mathbf{t}_F, \boldsymbol{\varphi}_F)^T$ des mobilen Fahrzeugs ergibt die gesuchte Pose $(\mathbf{t}_{Obj}, \boldsymbol{\varphi}_{Obj})^T$ des Objekts gegenüber dem Start- bzw. Bezugspunkt (s. Gleichung 7.1). Nur Letztere findet Eingang in die Umgebungskarte.

$$\begin{pmatrix} \mathbf{t}_{Obj} \\ \boldsymbol{\varphi}_{Obj} \end{pmatrix} = \begin{pmatrix} \mathbf{t}_F \\ \boldsymbol{\varphi}_F \end{pmatrix} + \begin{pmatrix} \mathbf{t}_{F_Obj} \\ \boldsymbol{\varphi}_{F_Obj} \end{pmatrix} \quad \text{bzw.} \quad \begin{pmatrix} x_{Obj} \\ y_{Obj} \\ z_{Obj} \\ \gamma_{Obj} \\ \beta_{Obj} \\ \alpha_{Obj} \end{pmatrix} = \begin{pmatrix} x_F \\ y_F \\ z_F \\ \gamma_F \\ \beta_F \\ \alpha_F \end{pmatrix} + \begin{pmatrix} x_{F_Obj} \\ y_{F_Obj} \\ z_{F_Obj} \\ \gamma_{F_Obj} \\ \beta_{F_Obj} \\ \alpha_{F_Obj} \end{pmatrix} \quad (7.1)$$

Bei der Exploration und Kartengenerierung mit Hilfe des mobilen Fahrzeugs können folgende Bewegungsarten verwendet werden:

- Fernsteuerung (s. Kap. 7.1.1),
- Pose anfahren (s. Kap. 7.2.3),
- Zufallsfahren (s. Kap. 7.2.4),
- Pfadfahren ohne Karte (s. Kap. 7.2.6),
- Pfadfahren mittels eines geplanten Pfades (Kap. 7.2.6) anhand einer „alten“ Karte.

Sowohl bei der Objektidentifizierung als auch bei der Hinderniserkennung werden Objekte der Fahrzeugumgebung analysiert und entsprechend für die Kartengenerierung und Hindernisvermeidung verwendet. Abhängig vom Objekttyp müssen diese jedoch unterschiedlich berücksichtigt werden. In **Tab. 7.1** sind die Eigenschaften von Objekten und deren Berücksichtigungen für die Kartengenerierung und für die Hindernisvermeidung zusammengestellt. Demnach müssen ortsveränderliche bzw. sich bewegende Objekte zwar bei der Hindernisvermeidung (s. Kap. 7.1.2) beachtet werden. Sie dürfen jedoch nicht in einer Karte gespeichert werden. Ortsfeste Objekte hingegen finden in beiden Verfahren gleichermaßen ihre Berücksichtigung.

	stehende feste Objekte	stehende form- veränderliche Objekte	bewegliche feste Objekte	bewegliche form- veränderliche Objekte
Hindernis- vermeidung	X	X	X	X
Karten- generierung	X	X	---	---

Tab. 7.1: Berücksichtigung verschiedener Objekttypen bei der Hindernisvermeidung (s. Kap. 7.1.2) und bei der Kartengenerierung.

7.2.6 Pfad- bzw. Bahnplanung und Pfad- bzw. Bahnfahrt ohne/mit Karte

Ein Pfad bzw. eine Bahn kann beispielsweise mit Hilfe der in Kap. 4.3.5.e beschriebenen GUI zunächst virtuell erstellt werden. Innerhalb der GUI-Arbeitsfläche kann der Pfad mit vordefinierten Geraden- oder Kurvenstücke zusammengesetzt werden. Der resultierende Pfad wird hierbei ohne eine hinterlegte Umgebungskarte erstellt.

Wird nun in die GUI-Arbeitsfläche eine Umgebungskarte geladen, können beim Erstellen eines virtuellen Pfads die darin enthaltenen Hindernisse einbezogen werden. Pfad- / Bahnplanungs-Software, wie z.B. „Adept MobilePlanner“ [239], kann die Pfadlage und -länge optimieren und darüber hinaus auch die Fahrzeugeigenschaften (Fahrzeug-Kinematik), wie Fahrwerk, Dimensionierung, Schwerpunkt, usw. berücksichtigen. Der virtuelle Pfad wird schließlich in Wegpunkte transformiert, d.h. Startpunkt, mehrere Zwischenpunkte und Endpunkt. Diese Wegpunkte können dann an das mobile Fahrzeug übermittelt werden.

Innerhalb der Pfad- / Bahnregelung des mobilen Fahrzeugs werden diese Wegpunkte als Sollwerte von Positionen gehandhabt. Die aktuelle Position des mobilen Fahrzeugs selbst hingegen wird stets mit Sensoren gemessen und als Istwert interpretiert. Ein geeigneter Regelungsalgorithmus, z.B. ein PID-Regler, ermittelt die erforderliche Fahrrichtung und -geschwindigkeit und ermöglicht damit das Anfahren des neuen Wegpunktes mit entsprechend neuer Position.

Die einzunehmende Orientierung weist stets in Fahrrichtung des mobilen Fahrzeugs und wird durch das Anfahren des nächsten anvisierten Wegpunkts hervorgerufen. Die Orientierung stellt sich somit automatisch ein und braucht nicht explizit angegeben zu werden. Jedoch soll bei Bedarf mittels der teilautonomen Funktion „Orientierung einnehmen“ (s. Kap. 7.2.3) eine gewünschte Orientierung mitgeteilt werden können. Auf diese Weise wird mit jedem Wegpunkt nacheinander verfahren, sodass eine kontinuierliche Fahrbewegung mit dem mobilen Fahrzeug auf einem vorgegebenen Pfad entsteht.

7.2.7 Andocken ohne/mit Karte

Ein mobiles Fahrzeug soll auf einem Fahrweg teilautonom fahren und an einer Be- / Entladerampe teilautonom andocken können. Die Be- / Entladerampe ist mit einer exakt ausgemessenen künstlichen Referenzmarke versehen [84]. Wie in Kap. 4 beschrieben, verfügt das mobile Fahrzeug über eine PMD-Kamera und eine 2D-Farbkamera. Während der Fahrt erlaubt die PMD-Kamera eine Fahrweg- und Hinderniserkennung (Kap. 3.4), deren Daten innerhalb der Hindernisvermeidung weiterverarbeitet werden (Kap. 5.2). Mit Hilfe der 2D-Farbkamera kann die Referenzmarke anhand der Textur identifiziert werden. Darüber hinaus dient sie zur Bestimmung der Distanz und des Orientierungswinkels der Referenzmarke relativ zum Fahrzeug (Kap. 6.3).

Ein Andock-Vorgang gilt erst dann als abgeschlossen, wenn die Fahrzeugfront parallel zur Be- / Entladerampe liegt. Dies kann mit dem in Kap. 4.2.2 beschriebenen IR-Front-Bumper, aber auch mit der PMD-Kamera überprüft werden. Für einen präziseren Andock-Vorgang muss die optische Hauptachse der 2D-Farbkamera parallel zum Normalenvektor der Referenzmarke verlaufen. Drei Varianten des Andockens seien genannt:

1. Innerhalb der Umgebungskarte wird ein Pfad als Andock-Vorgang entworfen und dem mobilen Fahrzeug in Form von Pose-Daten zur Abarbeitung gegeben (Kap. 7.2.3). Die Referenzmarke dient lediglich zur Überprüfung der Aktionen.

2. Der Andock-Vorgang selbst wird als teilautonome Funktion implementiert, wobei die Selbstlokalisierung anhand der Referenzmarke sowie geeigneter Regelungsalgorithmen, wie z.B. Fuzzy-Logic-Regler [244], die zentralen Bestandteile bilden. Zu Beginn und während des gesamten Andock-Vorgangs wird stets die gewünschte Referenzmarke innerhalb der 2D-Videobildfolge der 2D-Farbkamera identifiziert und innerhalb der PMD-Videobildfolge die relative Pose gegenüber dem mobilen Fahrzeug bestimmt. Die Identifizierung, die Posebestimmung und die Fahrbewegungen werden kontinuierlich durchgeführt, sodass das mobile Fahrzeug auf optimalem Pfad an die Be- / Entladerampe heranfährt.

3. Die beiden zuvor genannten Varianten können auch als Mischform verwendet werden. Das mobile Fahrzeug fährt auf dem von der Pfadplanung vorgegebenen Pfad zwecks Durchführung einer Dienstleistung und gelangt so in die Nähe der Be- / Entladerampe bzw. deren Referenzmarke. Mit der Erfassung der Referenzmarke durch die bildgebende Sensorik kann jetzt die teilautonome Funktion „Andock-Vorgang“ des mobilen Fahrzeugs aktiviert werden, die nun die unter 2. beschriebene Feinarbeit erledigt.

7.2.8 Vollautonomie

Ein vollautonomes System eignet sich für Handlungen, die regelmäßig und in abgegrenzten Arealen durchgeführt werden können. Außer Start und Ende der Aktion soll dies ohne Mitwirkung des Operators geschehen. Dabei sind die in Kap. 1 beschriebenen Asimovschen Verhaltensregeln für Roboter einzuhalten. Für Notsituationen, in denen Personen zu Schaden kommen können, muss daher zusätzlich eine für jede Person zugängliche Notaus-Funktion verfügbar sein.

Ein solches System ist bereits mit vorhandenen Anwendungen realisierbar: Beispielsweise soll ein Rasenmäher-Roboter selbständig seine Ladestation aufsuchen können, um bei Bedarf die Akkus aufzuladen. Ansonsten verrichtet dieses System die Mäharbeiten zeitabhängig (z.B. je nach Jahreszeit 0-, 1- oder 2-mal pro Woche) und witterungsabhängig (z.B. nur bei Sonnenwetter).

8 Ausblick

8.1 Systematik (teil)autonomer Funktionen und Dienstleistungen

Brauchbare Dienstleistungen, wie Transport-, Such- und Bewachungsaufgaben können bereits mit Hilfe eines einfachen mobilen Fahrzeugs durchgeführt werden, welches als Aktuator nur den Fahrtrieb selbst besitzt. Allerdings muss der Aufwand an konventioneller und bildgebender Sensorik sowie Verarbeitungs-Software für diese Aufgaben stets angepasst sein.

Dienstleistende mobile Systeme werden in der Fachwelt auch Service-Roboter genannt. Ein Service-Roboter für Reinigungsaufgaben besitzt z.B. neben dem Fahrtrieb zusätzlich mindestens einen weiteren Aktuator, das Reinigungswerk, welches ein- / ausgeschaltet und ggf. gehoben/gesenkt werden muss. Dieses Beispiel zeigt, dass sich die Komplexität von Dienstleistungen eines Service-Roboters durch Integration von zusätzlichen Aktuatoren, einschließlich Sensorik und Software, steigern lässt.

Sinnvolle Erweiterungen von Aktuatoren münden schließlich in einem menschenähnlichen Service-Roboter, einem sog. Humanoiden, der auch einen Fahr- bzw. Radantrieb besitzen kann. Gegenüber einem Schreitantrieb (s. **Abb. 1.2.f**) benötigt der Radantrieb einen geringeren Hard- und Software- sowie Energieaufwand. Der geringste Aufwand wird mit einem Differentialantrieb als Fahrtrieb für einen humanoiden Service-Roboter erreicht, jedoch mit dem Nachteil, dass damit Kanten, Stufen oder Treppen nicht befahren werden können. In anderen Projekten mit humanoiden Service-Robotern, wie Care-O-Bot [9], Rob-at-Work [9], Justin [10] (**Abb. 1.2.c, d** und **e**), wurde dieser Typus von Fahrwerk erfolgreich eingesetzt. Als bildgebende Sensoren finden sich Stereovision und/oder 2D- sowie 3D-Laser-scanner.

Einen zukunftsweisenden Ansatz verfolgt das Lynkeus-Teilprojekt „Griff-in-die-Kiste“ [84], in dem zunächst nur ein einzelner Handhabungsroboterarm mit einer PMD-Kamera ausgestattet wurde und damit Greifoperationen implementiert werden konnten. Genau diese Verbindung von Service-Robotik und PMD-Kamera zur

schnellen 3D-Umgebungserfassung bilden den zentralen Kern für komplexe teilautonome Funktionen und Dienstleistungen.

So könnte das bereits in **Abb. 4.1** I. gezeigt mobile Fahrzeug mit Differentialantrieb zusätzlich mit einem oder gar zwei Handhabungsroboterarmen ausgestattet werden. Entsprechend **Abb. 8.1** entstünde so ein humanoider Service-Roboter mit Fährantrieb (s. a. **Abb. 1.2.c, d, e** und **Abb. 1.3** I. und M.).

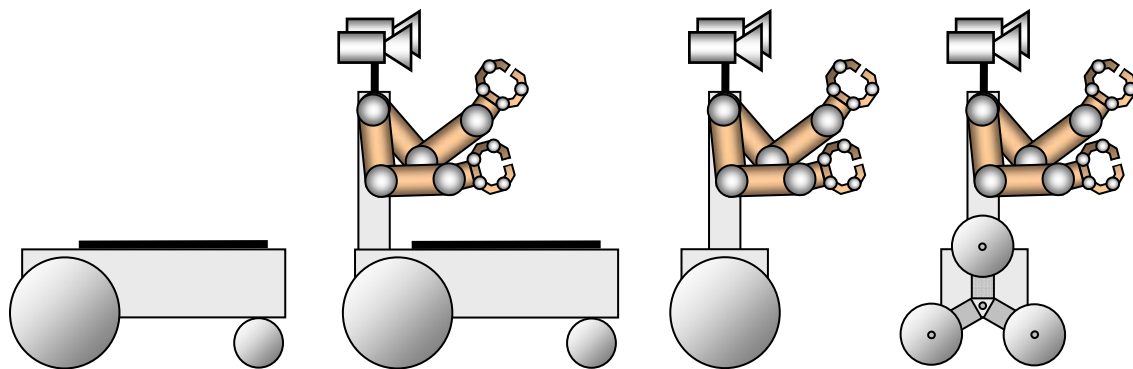


Abb. 8.1: Metamorphose einer a.) mobilen Fahrplattform hin zu einem b.) humanoiden Service-Roboter mit Differentialantrieb und zwei losen Rollen, bzw. c.) nur mit Differentialantrieb (ähnlich dem Segway-Fährantrieb [240]) und d.) mit Sternradantrieb zwecks Befahren von Treppenstufen.

Zukünftige Forschungsaktivitäten könnten die gezielte Realisierung von Dienstleistungen untersuchen, die in einem solchen Service-Roboter enthalten sein sollten. Es gilt zu prüfen, welche teilautonomen Funktionen nötig sind, um gewünschte Dienstleistungen zu ermöglichen. In **Tab. 8.1** werden teilautonome Funktionen sowie Dienstleistungen vorgeschlagen, die in einem humanoiden Service-Roboter implementiert sein und via Telepräsenz innerhalb einer geeigneten GUI per Tastatur, Computer-Maus und/oder Joystick durch den Operator aufgerufen werden können.

Des Weiteren stellt sich die Frage, in wieweit standardisierte Software einbezogen werden kann. Einen viel versprechenden Ansatz bietet hierfür die Fa. Microsoft mit ihrer Software MRDS (Microsoft Robotics Developer Studio) [241]. MRDS stellt eine Plattform mit einheitlichen Schnittstellen dar, die über eine universelle grafische Entwicklungsumgebung verfügt, zwecks Simulation und Ansteuerung von Robotersystemen. Software-Module können sowohl in fremden Programmiersprachen wie auch in „Microsoft Visual .NET“ oder sogar in der grafischen Programmiersprache VPL (Microsoft Visual Programming Language) programmiert und integriert werden.

		mobiles Fahrzeug	Zusatz- funktionen	Handhabungs- roboterarm	
Komplexität ↑ hoch ↓ gering	Globale Navigation	Dienstleistungen	Kurier- und Transportdienst, Einparken, Andocken, Objektsuche, Überwachung eines Objekts (Event Monitoring), Kartenaktualisierung	Selbstlokalisierung, Positions-, Pfad-, Bahnplanung mit Karte	gefundenes oder vom Operator ausgewähltes Objekt an optimaler Stelle ergreifen, Objekt am definierten Ort ablegen
			Exploration, Kartengenerierung	Selbstlokalisierung	
		Teilautonome Funktionen II	Pose anfahren, Pfad- / Bahnfahrt, letzte(n) Fahrabschnitt(e) zurückfahren (UnDo-Funktion)	Selbstlokalisierung, Positions-, Pfad-, Bahnplanung ohne Karte	Position anfahren, Bahnbewegung, letzte Bewegung(en) zurückbewegen (UnDo-Funktion)
			Treppe befahren (nur mit geeignetem Fahrwerk)		
	Teilautonome Funktionen I	Operator folgen, Referenzmarken folgen, Fahrweg folgen, Objekt folgen, Wand folgen, Bahnstreifen folgen	Objekterkennung		
		Zufallsfahrt, geometrische Figuren fahren, direkte Fernsteuerung (Vor-, Rück-, Links-, Rechtsfahren, Stopp, Links- / Rechtsdrehen)		direkte Fernsteuerung (Roboterarm u. Greifer heben, senken, vor, zurück, Greifer links und rechts drehen, mit Greifen zugreifen)	
		Hindernisvermeidung	Hindernis- und Fahrwegerkennung	Kollisionsvermeidung	
		Geschwindigkeitsregelung		Bewegungsregelung	

Tab. 8.1: Hierarchie einiger (teil)autonomer Funktionen und Dienstleistungen eines Service-Roboters.

8.2 Absolute Selbstlokalisierung mit Hilfe des WLAN

Bei den bisherigen Lösungen, wie z.B. Microsoft RADAR [245], Fraunhofer Navigator bzw. Awiloc [246], MagicMap [247], OpenWLANMap [248], usw. wird lediglich die empfangenen Feldstärken von mehreren WLAN-Access-Points ausgewertet. D.h. innerhalb eines WLAN-Sensors eines mobilen Fahrzeugs werden entsprechend der empfangenen Feldstärken die Distanzen zu den WLAN-Access-Points bestimmt. Mit Kenntnis der Positionen der jeweiligen WLAN-Access-Points und der ermittelten Distanzen kann daraus die Position des WLAN-Sensors bzw. des mobilen Fahrzeugs berechnet werden.

Da bei dieser Art der Selbstlokalisierung z.B. innerhalb von Gebäuden die Wände, Bewehrungen im Beton, Strom- und Wasserleitungen, usw. die Feldstärke beeinflussen, sind nur ungenaue Positionsberechnungen möglich. Werden statt der Distanzen die jeweiligen Positionswinkel der WLAN-Access-Points im Umkreis des WLAN-Sensors ermittelt, so könnte damit die Position des mobilen Fahrzeugs mit einer höheren Genauigkeit bestimmt werden.

Als Alternative zur WLAN-Technik mit dem Frequenzbereich 2,4 - ca. 5 GHz könnten andere Datenfunk-Geräte, z.B. im Frequenzbereich um 868 MHz für die Selbstlokalisierung eingesetzt werden, da bei niedrigeren Frequenzen die o.g. Störungen geringer ausfallen [253] [254].

9 Zusammenfassung

Diese Arbeit befasst sich mit der Hindernis- und Fahrwegerkennung, der Hindernisvermeidung sowie mit der Selbstlokalisierung eines mobilen Fahrzeugs. Während dafür bisher 2D-Laserscanner, 3D-Laserscanner oder Stereovision eingesetzt wurden, wird hier die Anwendung der PMD-Kamera gezeigt.

Als Experimentierplattform wurde ein mobiler Roboter mit Differentialantrieb (Tom3D) und ein weiterer mobiler Roboter mit Ackermannantrieb (Merlin3D) vorgestellt. Beide mobilen Fahrzeuge sind mit der gleichen Systemarchitektur ausgestattet und besitzen je einen Embedded-PC für Sensordatenverarbeitung, Bildverarbeitung und Kommunikation sowie einen Mikrocontroller C167 für Geschwindigkeitsregelung und Hindernisvermeidung. Für die Geschwindigkeitsregelung und für die grobe Selbstlokalisierung kommen Rad-Encoder zum Einsatz. GPS- und Inklinationsensoren dienen zur Unterstützung der Selbstlokalisierung. Der Zugriff auf das mobile Fahrzeug durch den Operator ist einerseits als manuelle Steuerung möglich oder kann andererseits per Wireless-LAN (WLAN) realisiert werden. Eine Telepräsenz bzw. Telemetrie via Mobilfunk oder Satellitenfunk ist denkbar.

Eine PMD-Kamera zusammen mit einer 2D-Kamera wurde mit einem definierten Neigungswinkel in das mobile Fahrzeug integriert. Dabei wurden der Eindeutigkeitsbereich (7,5 m), der optische Öffnungswinkel und die Montagehöhe der PMD-Kamera berücksichtigt. Aus dieser Perspektive nimmt die PMD-Kamera stets Hindernisse und Fahrweg des Frontbereichs des mobilen Fahrzeugs insgesamt auf. Eine Unterscheidung zwischen Hindernissen und Fahrweg innerhalb des PMD-Videobildes ist notwendig. Auch muss zwischen sog. negativen und positiven Hindernissen differenziert werden. Für eine echtzeitfähige Separierung von Fahrweg, negativen und positiven Hindernissen müssen bei jedem PMD-Videobild der Neigungswinkel der PMD-Kamera sowie der Pitch- und der Roll-Winkel des mobilen Fahrzeugs berücksichtigt werden.

Nach erfolgter Extrahierung der negativen und positiven Hindernisse wird das PMD-Videobild in gleich große vertikale Segmente unterteilt. Daraus wird das Segment (Segment-Nummer) ermittelt, welches die kleinste Distanz inne hat. Die Segment-

Nummer und der minimale Distanzwert stellen die linguistischen Eingangsvariablen zweier Fuzzy-Logic-Regler dar. Nach der Fuzzifizierung folgt die Fuzzy-Inferenz, in der die Fahrmanöver zur Hindernisvermeidung linguistisch formuliert sind. Die daraus resultierenden Ergebnisse werden in der Defuzzifizierung schließlich in konkrete Motorsteuerungssignale umgesetzt. Mit Hilfe dieser Schritte ist eine Navigation ohne Umgebungskarte und ohne Pfad- bzw. Bahnregelung realisierbar. Das mobile Fahrzeug kann damit bereits für ungesteuertes Fahren bzw. Zufallsfahren für Explorationsaufgaben genutzt werden.

Erst mit der Verfügbarkeit einer Selbstlokalisierung, hier speziell der relativen Selbstlokalisierung und Koppelnavigation, sind Pfad- / Bahnfahren und Kartengenerierung möglich. Fahrbewegungen eines mobilen Fahrzeugs können durch Lageänderungen von Objekten bzw. deren markanten Merkmalen innerhalb der PMD-Videobildfolge einer PMD-Kamera beobachtet werden. Diese Lageänderungen gilt es stets innerhalb zweier aufeinander folgender PMD-Videobilder zu bestimmen. Problematisch dabei ist jedoch, die markanten Merkmale des ersten PMD-Videobildes im zweiten PMD-Videobild wieder zu finden und einander zuzuordnen. Aus diesen Lageänderungen können dann Translationsvektor und Rotationsmatrix bzw. die Poseänderung (Position und Orientierung) des mobilen Fahrzeugs errechnet werden.

Markante Merkmale innerhalb der beteiligten PMD-Videobilder lassen sich u.a. mit Hilfe des sog. Moravec- bzw. Interest-Operators extrahieren. Die Korrespondenzbildung zwischen beiden Merkmalspunktmenge kann nun mit Hilfe der Euklidischen Distanz bzw. der Euklidischen Norm erfolgen. Auf Basis dieser Paarbildungen können mittels der Einheitsquaternionen-Methode für 3D-Bewegungen oder mittels der HAYAI (Highspeed And Yet Accurate Indoor/outdoor-tracking) -Methode für 2D-Bewegungen nun die gesuchte Rotationsmatrix und anschließend der Translationsvektor bestimmt werden. Translationsvektor und Rotationsmatrix weisen Fehler auf, die mit Hilfe der ICP (Iterative Closest/Corresponding Points) -Methode iterativ verringert werden können.

Einfache Fahrmanöver, Hinderniserkennung und -vermeidung, Selbstlokalisierung, Pfad- / Bahnfahren, usw. lassen sich miteinander kombinieren und somit teilautonome Funktionen kreieren. Eine hierarchische Zusammenstellung zeigt den

Übergang von reiner Fernsteuerung, über teilautonome und autonome Funktionen bis hin zu brauchbaren Dienstleistungen eines mobilen Fahrzeugs.

10 Anhang

10.1 Matrizen [195] [199]

10.1.1 Rotationsmatrizen

$$\mathbf{R}_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{pmatrix}, \quad \mathbf{R}_y(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix}, \quad \mathbf{R}_z(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (10.1)$$

$$\mathbf{R} = \mathbf{R}_z(\alpha) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_x(\gamma) \quad (10.2)$$

$$\mathbf{R} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \beta & \sin \beta \cdot \sin \gamma & \sin \beta \cdot \cos \gamma \\ 0 & \cos \gamma & -\sin \gamma \\ -\sin \beta & \cos \beta \cdot \sin \gamma & \cos \beta \cdot \cos \gamma \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} \cos \alpha \cdot \cos \beta & \cos \alpha \cdot \sin \beta \cdot \sin \gamma - \sin \alpha \cdot \cos \gamma & \cos \alpha \cdot \sin \beta \cdot \cos \gamma + \sin \alpha \cdot \sin \gamma \\ \sin \alpha \cdot \cos \beta & \sin \alpha \cdot \sin \beta \cdot \sin \gamma + \cos \alpha \cdot \cos \gamma & \sin \alpha \cdot \sin \beta \cdot \cos \gamma - \cos \alpha \cdot \sin \gamma \\ -\sin \beta & \cos \beta \cdot \sin \gamma & \cos \beta \cdot \cos \gamma \end{pmatrix}$$

10.1.2 Homogene Rotationsmatrizen

$$\mathbf{R}_{hx}(\gamma) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma & 0 \\ 0 & \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{R}_{hy}(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (10.3)$$

$$\mathbf{R}_{hz}(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

10.2 Multivariate Normalverteilung [199] - [212]

10.2.1 Empirischer arithmetischer Mittelwert

$$\bar{x}_n = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{und} \quad \bar{y}_n = \frac{1}{n} \sum_{i=1}^n y_i \quad (10.4)$$

10.2.2 Varianz und Kovarianz zweier Zufallsvariablen X und Y

$$\text{Kov}(X,X) = \sigma_{xx} = \sigma_x^2 = E((\mathbf{x} - \boldsymbol{\mu}_x) \cdot (\mathbf{x} - \boldsymbol{\mu}_x)) = E((\mathbf{x} - \boldsymbol{\mu}_x)^2) = \text{Var}(X) \quad (10.5)$$

$$\text{Kov}(Y,Y) = \sigma_{yy} = \sigma_y^2 = E((\mathbf{y} - \boldsymbol{\mu}_y) \cdot (\mathbf{y} - \boldsymbol{\mu}_y)) = E((\mathbf{y} - \boldsymbol{\mu}_y)^2) = \text{Var}(Y) \quad (10.6)$$

$$\text{Kov}(X,Y) = \sigma_{xy} = E((\mathbf{x} - \boldsymbol{\mu}_x) \cdot (\mathbf{y} - \boldsymbol{\mu}_y)) \quad (10.7)$$

$$\text{Kov}(Y,X) = \sigma_{yx} = E((\mathbf{y} - \boldsymbol{\mu}_y) \cdot (\mathbf{x} - \boldsymbol{\mu}_x)) \quad (10.8)$$

10.2.3 Empirische Varianz und Kovarianz zweier Zufallsvariablen X und Y

$$\text{Kov}_n(X,X) = \sigma_{xx} = \sigma_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x}_n) \cdot (x_i - \bar{x}_n) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x}_n)^2 = \text{Var}_n(X) \quad (10.9)$$

$$\text{Kov}_n(Y,Y) = \sigma_{yy} = \sigma_y^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y}_n) \cdot (y_i - \bar{y}_n) = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y}_n)^2 = \text{Var}_n(Y) \quad (10.10)$$

$$\text{Kov}_n(X,Y) = \sigma_{xy} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x}_n) \cdot (y_i - \bar{y}_n) \quad (10.11)$$

$$\text{Kov}_n(Y,X) = \sigma_{yx} = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y}_n) \cdot (x_i - \bar{x}_n) \quad (10.12)$$

10.2.4 Kovarianzmatrix zweier Zufallsvariablen X und Y

$$\boldsymbol{\Sigma}_{xy} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{pmatrix} \quad (10.13)$$

10.2.5 Korrelationskoeffizient zweier Zufallsvariablen X und Y

$$\rho(X,Y) = \frac{\text{Kov}(X,Y)}{\sqrt{\text{Var}(X)} \cdot \sqrt{\text{Var}(Y)}} \quad (10.14)$$

$$\rho_n(X,Y) = \frac{\text{Kov}_n(X,Y)}{\sqrt{\text{Var}_n(X)} \cdot \sqrt{\text{Var}_n(Y)}} \quad (10.15)$$

10.3 Transformation einer multivariaten Normalverteilung [96] [116] [117] [199] - [209]

10.3.1 Lineare Transformation

Die lineare Transformation einer multivariaten Normalverteilung erfolgt an linearen Übertragungsfunktionen $\mathbf{y}(\mathbf{x})$, die durch das Geradengleichungssystem (10.16) beschrieben werden, wobei \mathbf{A} eine $n \times m$ -Matrix sowie \mathbf{b} einen Vektor von Konstanten darstellen. Die Vektoren \mathbf{x} und \mathbf{y} bilden den Eingangs- und den Ausgangsvektor.

$$\mathbf{y}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x} + \mathbf{b} \quad (10.16)$$

Nach der linearen Transformation ergeben sich für die Erwartungswerte und für die dazu gehörende Kovarianzmatrix die Beziehungen in (10.17) und (10.18):

$$\begin{aligned}
 \boldsymbol{\mu}_y &= E(\mathbf{y}) \\
 \boldsymbol{\mu}_y &= E(\mathbf{A} \cdot \mathbf{x} + \mathbf{b}) \\
 \boldsymbol{\mu}_y &= \mathbf{A} \cdot E(\mathbf{x}) + \mathbf{b} \\
 \boldsymbol{\mu}_y &= \mathbf{A} \cdot \boldsymbol{\mu}_x + \mathbf{b} \tag{10.17}
 \end{aligned}$$

$$\begin{aligned}
 \boldsymbol{\Sigma}_y &= E((\mathbf{y} - E(\mathbf{y})) \cdot (\mathbf{y} - E(\mathbf{y}))^T) \\
 \boldsymbol{\Sigma}_y &= E((\mathbf{A} \cdot \mathbf{x} + \mathbf{b} - \mathbf{A} \cdot E(\mathbf{x}) - \mathbf{b}) \cdot (\mathbf{A} \cdot \mathbf{x} + \mathbf{b} - \mathbf{A} \cdot E(\mathbf{x}) - \mathbf{b})^T) \\
 \boldsymbol{\Sigma}_y &= E((\mathbf{A} \cdot \mathbf{x} - \mathbf{A} \cdot E(\mathbf{x})) \cdot (\mathbf{A} \cdot \mathbf{x} - \mathbf{A} \cdot E(\mathbf{x}))^T) \\
 \boldsymbol{\Sigma}_y &= E(\mathbf{A} \cdot (\mathbf{x} - E(\mathbf{x})) \cdot (\mathbf{x} - E(\mathbf{x}))^T \cdot \mathbf{A}^T) \\
 \boldsymbol{\Sigma}_y &= \mathbf{A} \cdot E((\mathbf{x} - E(\mathbf{x})) \cdot (\mathbf{x} - E(\mathbf{x}))^T) \cdot \mathbf{A}^T \\
 \boldsymbol{\Sigma}_y &= \mathbf{A} \cdot \boldsymbol{\Sigma}_x \cdot \mathbf{A}^T \tag{10.18}
 \end{aligned}$$

10.3.2 Nichtlineare Transformation

Bei der nichtlinearen Transformation (s. **Abb. 10.1**) einer multivariaten Normalverteilung ist eine Linearisierung der zunächst nichtlinearen Übertragungsfunktionen $\mathbf{y}(\mathbf{x})$ erforderlich, indem diese mittels einer Taylor-Reihe (10.19) entwickelt und alle Elemente ab dem dritten Term (10.20) vernachlässigt werden:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\boldsymbol{\mu}_x) + \mathbf{f}'(\boldsymbol{\mu}_x) \cdot (\mathbf{x} - \boldsymbol{\mu}_x) + \frac{1}{2!} \mathbf{f}''(\boldsymbol{\mu}_x) \cdot (\mathbf{x} - \boldsymbol{\mu}_x)^2 + \frac{1}{3!} \mathbf{f}'''(\boldsymbol{\mu}_x) \cdot (\mathbf{x} - \boldsymbol{\mu}_x)^3 + \dots \tag{10.19}$$

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{y} = \mathbf{f}(\boldsymbol{\mu}_x) + \mathbf{f}'(\boldsymbol{\mu}_x) (\mathbf{x} - \boldsymbol{\mu}_x) \tag{10.20}$$

$$\begin{aligned}
 \rightarrow \mathbf{y} &= \mathbf{f}(\boldsymbol{\mu}_x) + \mathbf{f}'(\boldsymbol{\mu}_x) \cdot \mathbf{x} - \mathbf{f}'(\boldsymbol{\mu}_x) \cdot \boldsymbol{\mu}_x \\
 \mathbf{y} &= \mathbf{f}'(\boldsymbol{\mu}_x) \cdot \mathbf{x} + \mathbf{f}(\boldsymbol{\mu}_x) - \mathbf{f}'(\boldsymbol{\mu}_x) \cdot \boldsymbol{\mu}_x \\
 \mathbf{y} &= \nabla_x \mathbf{f}(\boldsymbol{\mu}_x) \cdot \mathbf{x} + \mathbf{f}(\boldsymbol{\mu}_x) - \nabla_x \mathbf{f}(\boldsymbol{\mu}_x) \cdot \boldsymbol{\mu}_x \tag{10.21}
 \end{aligned}$$

Ein Koeffizientenvergleich der linearisierten Transformationsfunktion (10.21) mit (10.16) ergibt $\mathbf{A} = \nabla_x \mathbf{f}(\boldsymbol{\mu}_x)$ und $\mathbf{b} = \mathbf{f}(\boldsymbol{\mu}_x) - \nabla_x \mathbf{f}(\boldsymbol{\mu}_x) \cdot \boldsymbol{\mu}_x$. Durch Einsetzen von \mathbf{A} und \mathbf{b} in die Gleichungen (10.17) und (10.18) erhält man die Erwartungswerte (10.22) und die dazu gehörende Kovarianzmatrix (10.23) einer nichtlinearen Transformation:

$$\begin{aligned}\boldsymbol{\mu}_y &= \mathbf{A} \boldsymbol{\mu}_x + \mathbf{b} \\ \boldsymbol{\mu}_y &= \nabla_x \mathbf{f}(\boldsymbol{\mu}_x) \cdot \boldsymbol{\mu}_x + \mathbf{f}(\boldsymbol{\mu}_x) - \nabla_x \mathbf{f}(\boldsymbol{\mu}_x) \cdot \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y &= \mathbf{f}(\boldsymbol{\mu}_x)\end{aligned}\quad (10.22)$$

$$\begin{aligned}\boldsymbol{\Sigma}_y &= \mathbf{A} \cdot \boldsymbol{\Sigma}_x \cdot \mathbf{A}^T \\ \boldsymbol{\Sigma}_y &= \nabla_x \mathbf{f}(\boldsymbol{\mu}_x) \cdot \boldsymbol{\Sigma}_x \cdot \nabla_x \mathbf{f}(\boldsymbol{\mu}_x)^T\end{aligned}\quad (10.23)$$

$$\text{mit } \nabla_x = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3}, \dots \right) \quad (10.24)$$

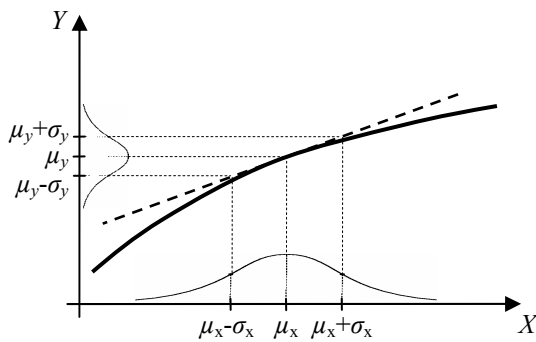


Abb. 10.1: Linearisierung einer nichtlinearen Transformation einer 1-dimensionalen Normalverteilung [96].

10.3.3 Funktion der Pose bzw. der Poseänderung

$$\mathbf{p}_{v+1} = \mathbf{f}(x_v, y_v, \alpha_v, \Delta s_v, \Delta \alpha_v) = \mathbf{p}_v + \Delta \mathbf{p}_v = \begin{pmatrix} x_v + \Delta s_v \cdot \cos\left(\alpha_v + \frac{\Delta \alpha_v}{2}\right) \\ y_v + \Delta s_v \cdot \sin\left(\alpha_v + \frac{\Delta \alpha_v}{2}\right) \\ \alpha_v + \Delta \alpha_v \end{pmatrix} \quad (10.25)$$

10.3.4 Kovarianzmatrix $\boldsymbol{\Sigma}_{v+1}$

Die aus den Gleichungen (10.22) und (10.23) hergeleitete Kovarianzmatrix $\boldsymbol{\Sigma}_{v+1} = \nabla_v \mathbf{f}(\boldsymbol{\mu}_v) \cdot \boldsymbol{\Sigma}_v \cdot \nabla_v \mathbf{f}(\boldsymbol{\mu}_v)^T + \nabla_{\Delta v} \mathbf{f}(\boldsymbol{\mu}_{\Delta v}) \cdot \boldsymbol{\Sigma}_{\Delta v} \cdot \nabla_{\Delta v} \mathbf{f}(\boldsymbol{\mu}_{\Delta v})^T$ in (6.26) benötigt für deren Lösung folgende Elemente:

$$\mathbf{f}(\boldsymbol{\mu}_v) = \mathbf{f}(\mu_{x_v}, \mu_{y_v}, \mu_{\alpha_v}) = \begin{pmatrix} \mu_{x_v} + \Delta s_v \cdot \cos\left(\mu_{\alpha_v} + \frac{\Delta \alpha_v}{2}\right) \\ \mu_{y_v} + \Delta s_v \cdot \sin\left(\mu_{\alpha_v} + \frac{\Delta \alpha_v}{2}\right) \\ \mu_{\alpha_v} + \Delta \alpha_v \end{pmatrix} \quad (10.26a)$$

$$\mathbf{f}(\boldsymbol{\mu}_{\Delta v}) = \mathbf{f}(\mu_{\Delta s_v}, \mu_{\Delta \alpha_v}) = \begin{pmatrix} x_v + \boldsymbol{\mu}_{\Delta s_v} \cdot \cos\left(\alpha_v + \frac{\boldsymbol{\mu}_{\Delta \alpha_v}}{2}\right) \\ y_v + \boldsymbol{\mu}_{\Delta s_v} \cdot \sin\left(\alpha_v + \frac{\boldsymbol{\mu}_{\Delta \alpha_v}}{2}\right) \\ \alpha_v + \boldsymbol{\mu}_{\Delta \alpha_v} \end{pmatrix} \quad (10.26b)$$

$$\nabla_{\Delta v} = \left(\frac{\partial}{\partial \Delta s_v}, \frac{\partial}{\partial \Delta \alpha_v} \right) \quad \text{und} \quad \nabla_v = \left(\frac{\partial}{\partial x_v}, \frac{\partial}{\partial y_v}, \frac{\partial}{\partial \alpha_v} \right) \quad (10.26c)$$

10.4 Quaternionen [29] [199] [219] - [229]

Eine Quaternion beschreibt eine Drehung mit dem Winkel φ um die Rotationsachse \mathbf{q} bzw. um den Einheitsvektor \mathbf{u} (s. **Abb. 10.2**).

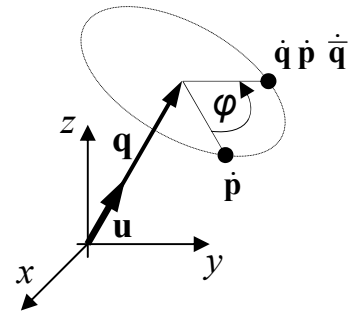


Abb. 10.2: Rotation eines Punktes \mathbf{p} mit Quaternionen.

$$\hat{\mathbf{q}} = \begin{pmatrix} \cos \varphi/2 \\ u_x \cdot \sin \varphi/2 \\ u_y \cdot \sin \varphi/2 \\ u_z \cdot \sin \varphi/2 \end{pmatrix} = \begin{pmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{pmatrix} = \begin{pmatrix} q_0 \\ \mathbf{q} \end{pmatrix} \quad (10.27)$$

Einheitsvektor \mathbf{u} der Quaternion $\hat{\mathbf{q}}$

$$\mathbf{u} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = \frac{\mathbf{q}}{|\mathbf{q}|} = \frac{\mathbf{q}}{\sin \varphi/2} \quad (10.28)$$

Quaternion in komplexer Darstellung

$$\hat{\mathbf{q}} = q_0 + \mathbf{i} q_x + \mathbf{j} q_y + \mathbf{k} q_z \quad (10.29a)$$

$$\text{mit } \mathbf{i}\mathbf{j} = -\mathbf{j}\mathbf{i} = \mathbf{k}, \quad \mathbf{j}\mathbf{k} = -\mathbf{k}\mathbf{j} = \mathbf{i}, \quad \mathbf{k}\mathbf{i} = -\mathbf{i}\mathbf{k} = \mathbf{j}, \quad \mathbf{i}^2 = \mathbf{k}^2 = \mathbf{j}^2 = -1 \quad (10.29b)$$

Quaternion als Vektor und als Matrix

$$\hat{\mathbf{q}} = \begin{pmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{pmatrix}, \quad \hat{\mathbf{Q}} = \begin{pmatrix} q_0 & -q_x & -q_y & q_z \\ q_x & q_0 & -q_z & -q_y \\ q_y & q_z & q_0 & q_x \\ -q_z & q_y & -q_x & q_0 \end{pmatrix} \quad (10.30)$$

Konjugierte Quaternion als Vektor und als Matrix [199]

$$\dot{\mathbf{q}} = \begin{pmatrix} q_0 \\ -q_x \\ -q_y \\ -q_z \end{pmatrix}, \quad \dot{\mathbf{Q}} = \begin{pmatrix} q_0 & q_x & q_y & -q_z \\ -q_x & q_0 & q_z & q_y \\ -q_y & -q_z & q_0 & -q_x \\ q_z & -q_y & q_x & q_0 \end{pmatrix} \quad (10.31)$$

Betrag einer Quaternion

$$|\dot{\mathbf{q}}| = \sqrt{q_0^2 + q_x^2 + q_y^2 + q_z^2} \quad (10.32a)$$

$$\text{für eine Einheitsquaternion gilt: } |\dot{\mathbf{q}}| = 1 \quad (10.32b)$$

Norm einer Quaternion

$$N(\dot{\mathbf{q}}) = 1 \quad (10.33a)$$

$$N^2(\dot{\mathbf{q}}) = \dot{\mathbf{q}} \cdot \dot{\mathbf{q}} = q_0^2 + q_x^2 + q_y^2 + q_z^2 = |\dot{\mathbf{q}}|^2 = 1 \quad (10.33b)$$

Multiplikation zweier Quaternionen

$$\dot{\mathbf{q}}_1 \cdot \dot{\mathbf{q}}_2 = (q_{01} + \mathbf{i} q_{x1} + \mathbf{j} q_{y1} + \mathbf{k} q_{z1}) \cdot (q_{02} + \mathbf{i} q_{x2} + \mathbf{j} q_{y2} + \mathbf{k} q_{z2}) \quad (10.34a)$$

$$= (q_{01}q_{02} - q_{x1}q_{x2} - q_{y1}q_{y2} - q_{z1}q_{z2}) + \mathbf{i} (q_{01}q_{x2} + q_{x1}q_{02} + q_{y1}q_{z2} - q_{z1}q_{y2}) + \\ + \mathbf{j} (q_{01}q_{y2} + q_{y1}q_{02} + q_{z1}q_{x2} - q_{x1}q_{z2}) + \mathbf{k} (q_{01}q_{z2} + q_{z1}q_{02} + q_{x1}q_{y2} - q_{y1}q_{x2})$$

$$\dot{\mathbf{q}}_1 \cdot \dot{\mathbf{q}}_2 = q_{01}q_{02} - \mathbf{q}_1 \cdot \mathbf{q}_2 + \mathbf{q}_1 \times \mathbf{q}_2 \quad (10.34b)$$

für Einheitsquaternionen gilt:

$$\dot{\mathbf{q}} \cdot \dot{\mathbf{q}} = \dot{\mathbf{q}} \cdot \dot{\mathbf{q}}^{-1} = \dot{\mathbf{q}} \cdot \dot{\mathbf{q}}^{-1} = 1 \quad (10.34c)$$

Inverse Quaternion

$$\dot{\mathbf{q}}^{-1} = \frac{\dot{\mathbf{q}}}{|\dot{\mathbf{q}}|^2} \quad (10.35a)$$

$$\text{für eine Einheitsquaternion gilt: } \dot{\mathbf{q}}^{-1} = \dot{\mathbf{q}} \quad (10.35b)$$

Polarform einer Quaternion

$$\dot{\mathbf{q}} = |\dot{\mathbf{q}}| \cdot \cos\left(\frac{\varphi}{2}\right) + \mathbf{i} \cdot |\dot{\mathbf{q}}| \cdot \sin\left(\frac{\varphi}{2}\right) + \mathbf{j} \cdot |\dot{\mathbf{q}}| \cdot \sin\left(\frac{\varphi}{2}\right) + \mathbf{k} \cdot |\dot{\mathbf{q}}| \cdot \sin\left(\frac{\varphi}{2}\right) \quad (10.36)$$

Rotation mit Hilfe einer Einheitsquaternion [219]

$$\dot{\mathbf{p}}_2 = \dot{\mathbf{q}} \dot{\mathbf{p}}_1 \dot{\mathbf{q}} \quad (10.37)$$

Nach [29] [199] [219] kann eine Rotation mit Hilfe von Einheitsquaternionen mit nachstehender Umformung erfolgen (s. **Abb. 10.2**):

$$\dot{\mathbf{p}}_2 = \dot{\mathbf{q}} \dot{\mathbf{p}}_1 \dot{\mathbf{q}} \quad (10.38)$$

$$\dot{\mathbf{p}}_2 = (\dot{\mathbf{Q}} \dot{\mathbf{p}}_1) \dot{\mathbf{q}}$$

$$\dot{\mathbf{p}}_2 = \dot{\mathbf{Q}}^T (\dot{\mathbf{Q}} \dot{\mathbf{p}}_1)$$

$$\dot{\mathbf{p}}_2 = (\dot{\mathbf{Q}}^T \dot{\mathbf{Q}}) \dot{\mathbf{p}}_1 \quad (10.39)$$

mit $\dot{\mathbf{p}}_1 = \begin{pmatrix} 0 \\ \mathbf{p}_1 \end{pmatrix}$, $\dot{\mathbf{p}}_2 = \begin{pmatrix} 0 \\ \mathbf{p}_2 \end{pmatrix}$ und

$$\dot{\mathbf{Q}}^T \dot{\mathbf{Q}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_z q_0) & 2(q_x q_z + q_y q_0) \\ 0 & 2(q_x q_y + q_z q_0) & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_x q_0) \\ 0 & 2(q_x q_z - q_y q_0) & 2(q_y q_z + q_x q_0) & q_0^2 - q_x^2 - q_y^2 + q_z^2 \end{pmatrix} = \left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & & & \\ 0 & \mathbf{R} & & \\ 0 & & & \end{array} \right)$$

$$\begin{pmatrix} 0 \\ \mathbf{p}_2 \end{pmatrix} = \left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & & & \\ 0 & \mathbf{R} & & \\ 0 & & & \end{array} \right) \cdot \begin{pmatrix} 0 \\ \mathbf{p}_1 \end{pmatrix} \quad (10.40)$$

$$\mathbf{p}_2 = \mathbf{R} \cdot \mathbf{p}_1 \quad (10.41)$$

mit

$$\mathbf{R} = \begin{pmatrix} q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_z q_0) & 2(q_x q_z + q_y q_0) \\ 2(q_x q_y + q_z q_0) & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_x q_0) \\ 2(q_x q_z - q_y q_0) & 2(q_y q_z + q_x q_0) & q_0^2 - q_x^2 - q_y^2 + q_z^2 \end{pmatrix} \quad (10.42a)$$

$$= \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (10.42b)$$

$$= \begin{pmatrix} \cos \alpha \cdot \cos \beta & \cos \alpha \cdot \sin \beta \cdot \sin \gamma - \sin \alpha \cdot \cos \gamma & \cos \alpha \cdot \sin \beta \cdot \cos \gamma + \sin \alpha \cdot \sin \gamma \\ \sin \alpha \cdot \cos \beta & \sin \alpha \cdot \sin \beta \cdot \sin \gamma + \cos \alpha \cdot \cos \gamma & \sin \alpha \cdot \sin \beta \cdot \cos \gamma - \cos \alpha \cdot \sin \gamma \\ -\sin \beta & \cos \beta \cdot \sin \gamma & \cos \beta \cdot \cos \gamma \end{pmatrix} \quad (10.42c)$$

Die Eulerschen Rotationswinkel können durch Vergleich der Matrix-Elemente der Matrizen in (10.42a) (10.42b) und (10.42c) wie folgt bestimmt werden:

$$\tan \alpha = \frac{r_{21}}{r_{11}} = \frac{\sin \alpha \cdot \cos \beta}{\cos \alpha \cdot \cos \beta} = \frac{2(q_x q_y + q_z q_0)}{q_0^2 + q_x^2 - q_y^2 - q_z^2} \quad (10.43a)$$

$$\sin \beta = -r_{31} = -2(q_x q_z - q_y q_0) \quad (10.43b)$$

$$\tan \gamma = \frac{r_{32}}{r_{33}} = \frac{\cos \beta \cdot \sin \gamma}{\cos \beta \cdot \cos \gamma} = \frac{2(q_y q_z + q_x q_0)}{q_0^2 - q_x^2 - q_y^2 + q_z^2} \quad (10.43c)$$

11 Literatur und Verweise

- [1] Asimov, Isaac: I, robot. New York: Gnome Press, c1950., 1st ed., 253 p. ; 21 cm, Contents: Introduction.- Robbie.- Run-around.- Reason.- Catch that rabbit.- Liar!- Little lost robot.- Escape!- Evidence.- The evitable conflict. <http://lccn.loc.gov/51009134>
- [2] Isaac Asimov: Robots and Empire. Garden City, N.Y.: Doubleday, 1st ed., ISBN 0-385-19092-1, 383 p, 1985.
- [3] Navigationssysteme für Fahrerlose Transportfahrzeuge (FTF), Fa. Götting KG, Lehrte: www.goetting.de
- [4] Staubsaugroboter „Robocleaner RC3000“ der Fa. Kärcher: www.kaercher.de
- [5] Staubsaugroboter „Roomba 560“ und Mobiler Roboter „PackBot“ der Fa. iRobot: www.iRobot.com
- [6] Rasenmäherroboter „Robomow RM 200“ der Fa. Friendly Robotics: www.friendlyrobotics.de
- [7] Rasenmäherroboter „Automower 230 ACX“ der Fa. Husqvarna: www.husqvarna.com
- [8] Mobile Roboter „Mosro (MOBiles Sicherheits-ROboter-System)“ und „Asendro-Scout“ der Fa. Robowatch Technologies: www.robowatch.de
- [9] Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart: www.ipa.fraunhofer.de
- [10] Deutsches Zentrum für Luft- und Raumfahrt, DLR-Oberpfaffenhofen, Institut für Robotik und Mechatronik: www.robotic.dlr.de
- [11] Humanoider Roboter „Asimo“ der Fa. Honda: <http://world.honda.com/ASIMO> oder <http://asimo.honda.com>
- [12] Ichbah, Dania: Roboter: Geschichte - Technik - Entwicklung / Daniel Ichbah. - Dt. Erstausg. . - München: Knesbeck, 2005. - 539 S.: zahlr. Ill., EST: Robots <dt.>. - Aus dem Franz. übers., ISBN: 3-89660-276-4, ISBN: 978-3-89660-276-3.
- [13] Schraft, Rolf Dieter: Service-Roboter-Visionen / Rolf Dieter Schraft ... - München [u.a.]: Hanser, 2004. - 223 S.: Ill., graph. Darst., ISBN: 3-446-22840-3.
- [14] Dossier: Roboter erobern den Alltag. Spektrum der Wissenschaft, Dossier 4/1998, ISSN 0947-7934: www.spektrum.de
- [15] Mobiler Roboter „tEODor“ der Fa. Telerob: www.telerob.de
- [16] DARPA Grand Challenge 2005: archive.darpa.mil/grandchallenge05
- [17] DARPA Urban Challenge 2007: archive.darpa.mil/grandchallenge
- [18] Mars-Exploration-Roboter „Mars Pathfinder“: <http://marsprogram.jpl.nasa.gov/MPF>
- [19] Mars-Exploration-Roboter „Spirit“ und „Opportunity“: <http://marsprogram.jpl.nasa.gov/programmissions/missions/present/2003>
- [20] Mars-Exploration-Roboter „Curiosity“: <http://marsprogram.jpl.nasa.gov/msl>
- [21] Informationen über „Exomars“ der DLR: http://www.dlr.de/irs/desktopdefault.aspx/tabid-5960/10970_read-25034
- [22] Informationen über „Exomars“ der ESA: <http://exploration.esa.int>

- [23] Technisches 3D-Sehen auf der Basis der PMD-Technologie, DFG-Arbeitstreffen am 07.10.2004 an der Universität Siegen: www.3d-view.org
- [24] Schwarte, R.: Ein Jahrzehnt 3D-Bilderfassung - Zielsetzungen, Fortschritte, Zukunftsaspekte. In: Tagungsband DGZfP/GMA - Optische Formerfassung, VDI/VDE-GMA, Stuttgart 1999.
- [25] Laserscanner der Fa. Sick AG, Waldkirch, Dtl.: www.sick.de
- [26] Laserscanner der Fa. Schmersal Holding KG, Wuppertal, Dtl.: www.schmersal.de
- [27] Hartmut Surmann, Kai Lingemann, Andreas Nüchter, Joachim Hertzberg: Aufbau eines 3D-Laserscanners für autonome mobile Roboter. GMD Report 126, GMD - Forschungszentrum Informationstechnik GmbH, 2001, ISSN 1435-2702, ISBN 3-88457-974-6.
- [28] Andreas Nüchter: Autonome Exploration und Modellierung von 3D-Umgebungen. Universität Bonn, Diplomarbeit 2002.
- [29] Andreas Nüchter: Semantische dreidimensionale Karten für autonome mobile Roboter. Universität Bonn, Dissertation 2006.
- [30] Nüchter, A.; Lingemann, K.; Hertzberg, J.; Surmann, H.; Pervolz, K.; Hennig, M.; Tiruchinapalli, K.R.; Worst, R.; Christaller, T.: Mapping of rescue environments with kurt3d. Safety, Security and Rescue Robotics, Workshop, 2005 IEEE International, June 6-9, 2005, Page(s) 1-6.
- [31] Surmann, H.; Nüchter, A.; Lingemann, K.; Hertzberg, J.: Kurt3D - An Autonomous Mobile Robot for Modelling the World in 3D. In: ERCIM News, Nr. 55, S. 24-25, ISSN: 0926-4981, 2003.
- [32] Stereokamera der Fa. Tyzx: www.tyx.com
- [33] Stereokamera und Fish-Eye-Kamera der Fa. Point Grey Research: www.ptgrey.com
- [34] 3D- bzw. PMD-Kameras der Fa. PMDTec GmbH, Am Eichenhang 50, 57076 Siegen: www.pmdtec.de
- [35] ZEISS, NRW - Zentrum für Sensorsysteme der Universität Siegen: www.zess.uni-siegen.de
- [36] Seyed Eghbal Ghobadi, Omar Edmond Loepprich, Oliver Lottner, Farid Ahmadov, Klaus Hartmann, Wolfgang Weihs, Otmar Löffeld: Analysis of the Personnel Safety in a Man-Machine-Cooperation Using 2D/3D Images. Proceedings of the EURON/IARP International Workshop on Robotics for Risky Interventions and Surveillance of the Environment, Benicassim, Spain, 2008.
- [37] Strahlenteiler der Fa. Edmund Industrie Optik GmbH: www.edmundoptics.de
- [38] Strahlenteiler der Fa. Sill Optics GmbH & Co. KG, Wendelstein: www.silloptics.de
- [39] 3D- bzw. PMD-Kamera der Fa. ifm electronic GmbH, Essen, Dtl.: www.ifm-electronic.de
- [40] Z-Cam der Fa. 3DV Systems Ltd., Yokneam, Israel: www.3dvsystems.com
- [41] 3D-Kamera der Fa. Canesta, Inc., Sunnyvale, California, USA: www.canesta.com
- [42] 3D-Kamera der Fa. MESA, Schweiz: Swissranger 3D: www.swissranger.ch: www.mesa-imaging.ch (früher: Fa. CSEM: www.csem.ch).
- [43] 3D-Kamera der Fa. CSEM, Zürich: www.csem.ch
- [44] T. Oggier, R. Kaufmann, N. Blanc: 3D-Bilder für Biometrie und Sicherheitsanwendungen - Lichtlaufzeitmessung bringt hohe Auflösung. In: Laser+Photonik 04/2005, Seite 16-18, ISSN: 1610-3521.

- [45] Bildverarbeitungsbibliothek OpenCV (Intel's Open Source Computer Vision Library): <http://sourceforge.net/projects/opencvlibrary>
- [46] A. Kolb: Kalibrierung und 2D/3D Bildverarbeitung mit dem PMD-Sensor. PMD Vision Day, 2008, München, Germany, 18. Nov. 2008.
- [47] M. Lindner, A. Kolb, and T. Ringbeck: New insights into the calibration of TOF sensors. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Workshop on ToF Camera based Computer Vision (TOF-CV), pages 1–5, 2008.
- [48] M. Lindner and A. Kolb: Calibration of the intensity-related distance error of the PMD ToF camera. In: Proc. SPIE, Intelligent Robots and Computer Vision, volume 6764, page 67640W, 2007.
- [49] M. Lindner and A. Kolb: Lateral and depth calibration of pmd-distance sensors. In: Proc. Int. Symp. on Visual Computing, LNCS, pages 524–533. Springer, 2006.
- [50] Kamera-Kalibrierung mit der Toolbox von Jean Yves Bouguet: www.vision.caltech.edu/bouguetj/calib_doc
- [51] Mathematische Software MATLAB der Fa. Mathworks: www.mathworks.com
- [52] Matthias Wiedemann, Markus Sauer, Frauke Driewer, Klaus Schilling: Analysis and characterization of the PMD camera for application in mobile robotics. In: Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, Korea, July 6-11, 2008.
- [53] Software CamVis Pro 2.0, Instruction Manual, Graphical User Interface for 3D Cameras. Version: 2.0d, Created: Sept. 02, 2004, Changed: June 7, 2005, PMDTec GmbH.
- [54] MiniSDK 1.0, Programming Manual, For 3D time-of-flight cameras. Version: 1.0 Created: 10/May/2005 Changed: 02/September/2005, PMDTec GmbH.
- [55] M. Stommel and K.-D. Kuhnert: Fusion of stereo-camera and PMD-camera data for real-time suited precise 3D environment reconstruction. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4780–4785, October 9-15, 2006.
- [56] Schwarte, R.; Aller, I.; Baumgarten, V.; Bundschuh, B.; Graf, W.; Hartmann, K.; Loffeld, O.: Laserradar mit Impulslaufzeitmessung. In: Intelligente Sensorsysteme in der Fertigungstechnik, J. Rogos (Hrsg.), Springer-Verlag, Berlin Heidelberg New York London Paris Tokyo Hong Kong, 1989.
- [57] Schwarte, R.: Ein Jahrzehnt 3D-Bilderfassung - Zielsetzungen, Fortschritte, Zukunftsaspekte. In: Tagungsband DGZfP/GMA - Optische Formerfassung, VDI/VDE-GMA, Stuttgart 1999.
- [58] Z. Xu: Investigation of 3D-Imaging Systems based on Modulated Light and Optical RF-Interferometrie (ORFI). Dissertation, INV, Universität Siegen, 1999.
- [59] Schwarte R., Heinol G. H., Buxbaum B., Ringbeck T., Xu Z. und Hartmann K.: Principles of Three-dimensional Imaging Techniques, in: Handbook of Computer Vision and Applications (Ed. B. Jähne) Academic Press (1999).
- [60] Lange, Robert: 3D time-of-flight distance measurement with custom solid-state image sensors in CMOS, CCD technology. - XI, 206 S.: Ill., graph. Darst. - Siegen, Univ., Diss., 2000.
- [61] Buxbaum, Bernd: Optische Laufzeitentfernungsmessung und CDMA auf Basis der PMD-Technologie mittels phasenvariabler PN-Modulation. Siegen, Univ., Diss., 2001, ISBN: 3-8265-9805-9.
- [62] B. Buxbaum, H. Hess, T. Ringbeck, R. Schwarte: The Photonic Mixer Device: New perspectives in optical wireless communication. Workshop „Optical MEMS and Integrated Optics“, Dortmund 2001.

- [63] Ringbeck, Thorsten: Untersuchung opto-elektrischer Phasenregelkreise auf Basis von Photomischdetektoren hinsichtlich deren Anwendungspotential für die Photonik. Shaker-Verl., 2002. - V, 102 S., (ZESS-Forschungsberichte ; Nr. 16). - Zugl.: Siegen, Univ., Diss., 2001, ISBN: 3-8265-9789-3.
- [64] Schwarte, R.: Dynamic 3D-Vision. IEEE Int. Symp. on Electron Devices for Microwave and Opto-electronic Applications, EDMO 2001, Wien.
- [65] Zhang, Zhigang: Untersuchung und Charakterisierung von PMD (Photomischdetektor)-Strukturen und ihren Grundschaltungen. - VII, 171 S.: Ill., graph. Darst. - Siegen, Univ., Diss., 2003.
- [66] H. Roth, R. Schwarte, N. Ruangpayoongsak, J. Kuhle, M. Albrecht, M. Grothof, H. Heß: 3D Vision Based on PMD-Technology for Mobile Robots. In: Proceedings SPIE Aerosense conference 2003, Volume „Unmanned Ground Vehicle Technology“, Paper 5083-66, Orlando, Florida, 2003.
- [67] R. Schwarte, H. Roth, B. Buxbaum, Z. Xu, T. Ringbeck, H. Heß, M. Albrecht, M. Grothof, Z. Zhang, J. Frey, H. Kraft, T. Möller, J. Wahrburg, T. Gollewski: Miniatur-3D-Kamera für mobile Telerobotik und Telemanipulation. In: Telematik 2003, VDI-Berichte 1785, Fachtagung, Juni 2003, Siegen, ISBN 3-18-091785-7, S. 65 - 77.
- [68] Rudolf Schwarte, Zhigang Zhang, Bernd Buxbaum: Neue 3D-Bildsensoren für das Technische 3D-Sehen. Eingeladener Vortrag auf dem VDE Kongress „Ambient Intelligence“, 18. – 20.10.2004, Berlin.
- [69] Rudolf Schwarte: Real Time 3D-perception by TOF-echoing 3D-Video Cameras. In: Dynamic Perception. Herausgeber: H. H. Bulthoff & H. A. Mallot Uwe J. Ilg, IOS Press, ISBN: 9781586034801, October 2004, p. 217-226.
- [70] Rudolf Schwarte: Ultra-fast Analog and Digital OE-Multichannel Signal Processing using PMD/OEP-Technologies. Proceedings of the 10th Microoptics Conference (MOC) in Jena, 2004.
- [71] H. Kraft, J. Frey, T. Moeller, M. Albrecht, M. Grothof, B. Schink, H. Hess, B. Buxbaum: 3D-Camera of High 3D-Frame Rate, Depth-Resolution and Background Light Elimination Based on Improved PMD (Photonic Mixer Device)-Technologies. OPTO-Conf., 2004, Nürnberg.
- [72] R. Schwarte, Z. Zhang, M. Grothof, J. Frey, H. Kraft, T. Moeller, H. Hess, B. Buxbaum, T. Ringbeck, Z. Xu: OEP (Opto Electronic Processor) for Extremely Fast Multi-Channel Analog and Digital OE-Signal-Processing. Proceedings of the SODC (Third Joint Symposium on Opto- & Microelectronic Devices and Circuits) in Wuhan, China, 2004.
- [73] Schwarte, R.: Breakthrough in Fast 3D-Imaging Using PMD- and OEP-Technology. IMS Workshop in Duisburg, May 25-26 2004.
- [74] Albrecht, Martin: Untersuchung von Photogate-PMD-Sensoren hinsichtlich qualifizierender Charakterisierungsparameter und –methoden. - XX, 207 S.: Ill., Siegen, Univ., Diss., 2007.
- [75] Th. Ringbeck, B. Hagebecker: Dreidimensionale Objekterfassung in Echtzeit – PMD Kameras erfassen proPixel Distanz und Helligkeit mit Videoframerate. - In: Allgemeine Vermessungs-Nachrichten (AVN). - Heidelberg: Wichmann, ISSN 0002-5968, 07/2007.
- [76] Jähne, Bernd: Digitale Bildverarbeitung. Springer ISBN: 3-540-41260-3, XII, 618 S., 248 Abb., 1 CD - ROM, 5. überarbeitete und erweiterte Aufl., 2002.
- [77] Driewer, Frauke: Teleoperation Interfaces in Human-Robot Teams. ISBN/ISMN: 3-923959-54-0: 978-3-923959-54-9. Würzburg, Univ., Diss., 2009, X, 138 S., Würzburger Forschungsberichte in Robotik und Telematik ; 1, Ill., graph. Darst., 2009.
- [78] Ghobadi, S.; Loeprich, O.; Lottner, O.; Hartmann, K.; Löffeld, O.; Weihs, W.: „Analysis of the Personnel Safety in a Man-Machine-Cooperation Using 2D/3D Images“, IARP/EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance of the Environment 2008, e-Treballs d'informica i tecnologia, Benicàssim, Spain, 7.-8. January, 2008, pp. 59.

- [79] Lottner, O.; Hartmann, K.; Weihs, W.; Loffeld, O.: „Image Registration and Calibration aspects for a new 2D / 3D camera“, EOS Conference on Frontiers in Electronic Imaging, Munich, Germany, June 18.-19., 2007, pp. 80-81.
- [80] Hartmann, K.; Loffeld, O.; Ghobadi, S.E.; Peters, V.; Prasad, T.D.A.; Sluiter, A.; Weihs, W.; Lerch, T.; Lottner, O.: „Klassifizierungsaspekte bei der 3D-Szenenexploration mit einer neuen 2D/3D-Multichip-Kamera“, Autonome Mobile Systeme 2005, Berlin, Heidelberg, Springer Verlag, 2005, pp. 75-80.
- [81] R. Schwarte: Breakthrough in Multi-Channel Laser-Radar Technology providing Thousands of High-Sensitive Lidar Receivers on a Chip. Remote Sensing 11th SPIE Intern. Symposium Laser Radar Techniques, 13-16. Sept. 2004 Maspalomas, Gran Canaria Islands, Spain.
- [82] Alexander Prusak, Hubert Roth, Rudolf Schwarte: Application of 3D-PMD Video Cameras for Tasks in the Autonomous Mobile Robotics. 16th IFAC World Congress, Prague, Czech Republic, from July 4 to July 8, 2005.
- [83] DFG-Projekt "Dyn3D" (Dynamic 3D-view), u.a. mit dem Subprojekt "3DPoseMap" (3D pose estimation and 3D mapping with help of PMD-Camera): www.zess.uni-siegen.de/cms/front_content.php?idcat=127
- [84] BMBF-Projekt „Lynkeus“, 2006-2009, u.a. mit dem Subprojekten „Autonome, mobile, messende Systeme“ und „Griff-in-die-Kiste“: www.lynkeus-3d.de
- [85] Pan-Tilt-Unit der Fa. Directed Perception, Inc., 890C Cowan Road, Burlingame, CA 94010: www.dperception.com
- [86] Andreas Knauf: Mathematische Physik: Klassische Mechanik. Springer, Berlin, 2011, ISBN 3-642-20977-7, ISBN 978-3-642-20977-2.
- [87] Hering, Ekbert: Physik für Ingenieure: mit 102 Tabellen / Ekbert Hering ; Rolf Martin ; Martin Stohrer. - 8. Aufl. - Berlin [u.a.]: Springer, 2002. - XXIII, 744 S.: Ill., graph. Darst. - (Springer-Lehrbuch), ISBN: 3-540-42964-6.
- [88] Kneubühl, Fritz Kurt: Repetitorium der Physik. - 5., überarb. Aufl. - Stuttgart: Teubner, 1994. - 544 S.: zahlr. graph. Darst. - (Teubner Studienbücher: Physik), ISBN: 3-519-43012-6.
- [89] Klaus Schilling, Hubert Roth: Vom Mars-Rover MIDD zu Anwendungen in Industrie und Ausbildung: Technologietransfer durch angewandte Forschung an der FH Ravensburg-Weingarten. In: horizonte 15 Internet-Ausgabe / November 1999, ISSN: 1432-9174.
- [90] A. Prusak, I. Schiller, O. Melnychuk, R. Koch, and H. Roth: Pose Estimation and Map Building with a PMD-Camera for Robot Navigation. In Dynamic 3D Imaging, Workshop in Conjunction with DAGM 2007, Heidelberg in Germany, September, 11th 2007.
- [91] A. Prusak, J. Bernshausen, H.-H. Götting, T. Neugebauer, H. Roth, J. Wahrburg: Applications of Automated Guided Vehicle (AGV) and Industry Robots with PMD-Camera. The 13th IASTED Intern. Conference on Robotics and Applications, ~RA 2007~, August 29 – 31, 2007.
- [92] Hirche, S., M. Buss: Study of Teleoperation Using Realtime Communication Network Emulation. IEEE/ASME International Conference on Advanced Intelligent Mechatronic AIM2003.
- [93] Janabi-Sharifi, F.: Collision: modelling, simulation and identification of robotic manipulators interacting with environments. Journal of Intelligent and Robotic Systems, 1995.
- [94] El-Kebir Boukas, Zi-Kuan Liu: Deterministic and Stochastic Time Delay Systems. Control Engineering, Birkhaeuser, 2002.
- [95] O. Roesch, H. Roth, A. Iqbal: Extended Stability margins on controller Design for Nonlinear Input Delay Systems. IFAC World-Congress, Prague, 2005.

- [96] Siegart, Roland; Nourbakhsh, Illah Reza: Introduction to autonomous mobile robots. Cambridge, Mass. [u.a.], MIT Press, 2004, IX, 321 S.: Ill., graph. Darst., ISBN 978-0-262-19502-7.
- [97] Mayr R., Regelungsstrategien für die automatische Fahrzeugführung. Längs- u. Querregelung; Spurwechsel- u. Überholmanöver, Springer-Verl., 178 S.; ISBN 3-540-67518-3.
- [98] Berufsgenossenschaften (BGS): www.dguv.de
(s. a. de.wikipedia.org/wiki/Berufsgenossenschaft)
- [99] Unfallverhütungsvorschrift Flurförderzeuge (BGV D27 bisherige UVV 48/VBG 36), Berufsgenossenschaft Handel und Warendistribution (BGHW):
http://medien-e.bghw.de/pdf/uvv_36.pdf
- [100] Prüfung von Flurförderzeugen (U D27.01), Berufsgenossenschaft Handel und Warendistribution (BGHW): http://medien-e.bghw.de/bge/pdf/u_d27_01.pdf
- [101] Betriebsanweisung für Flurförderzeuge (U D27.09), Berufsgenossenschaft Handel und Warendistribution (BGHW): http://medien-e.bghw.de/bge/pdf/u_d27_09.pdf
- [102] Deutsches Institut für Normung e.V. (DIN): www.din.de
- [103] DIN EN 1525 Sicherheit von Flurförderzeugen - Fahrerlose Flurförderzeuge und ihre Systeme, Beuth-Verlag, Berlin: www.beuth.de/de/norm/din-en-1525/3570013
- [104] Technischer Überwachungsverein (TÜV): www.tuev.de
- [105] Verein deutscher Ingenieure (VDI): www.vdi.de
- [106] VDI-Richtlinie 2510 Fahrerlose Transportsysteme:
[www.vdi.de/index.php?id=401&tx_vdirili_pi2\[showBeuthID\]=78228504](http://www.vdi.de/index.php?id=401&tx_vdirili_pi2[showBeuthID]=78228504)
- [107] VDI-Richtlinie 4451, Blatt 7 Kompatibilität von Fahrerlosen Transportsystemen, Leitsteuerung für FTS: www.vdi.de/46447.0.html?&tx_vdirili_pi2%5BshowUID%5D=91376
- [108] Günter Ullrich: Fahrerlose Transportsysteme: Eine Fibel – mit Praxisanwendungen – zur Technik – für die Planung / von Günter Ullrich . - Wiesbaden : Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH, Wiesbaden, ISBN: 9783834898906 , 2011.
(s. a. de.wikipedia.org/wiki/Fahrerloses_Transportfahrzeug)
- [109] Verein deutscher Sachversicherer (VdS): www.vds.de
- [110] IR-Sensoren „GP2D12“ und „GP2D120“ der Fa. Sharp Eletronic Components Group:
www.sharp.co.jp/ecg
- [111] Mobiler Roboter mit Rocker-Bogie-Fahrwerk: VolksBot-XT der Fa. Volksbot: www.volksbot.de
- [112] Peter Schrammel. Der Mars Pathfinder Rover Sojourner: Aufbau und Funktionalität eines mobilen Roboters. Robotik WS2002/03, Institut für Informationssysteme/Knowledge-Based Systems Group, Technische Universität Wien, 2002.
- [113] L. Richter: ExoMars – Neue Räder braucht der Mars. In: DRL Nachrichten: Magazin d. Dt. Zentrums für Luft- und Raumfahrt / DLR. - Köln: DLR 87.1997(Nov.) - ISSN: 0937-0420, S. 58-63.
- [114] Tietze, Ulrich: Halbleiter-Schaltungstechnik / Ulrich Tietze ; Christoph Schenk ; Eberhard Gamm . - 13., neu bearb. Aufl. - Heidelberg [u.a.]: Springer, 2010. - XXXVI, 1711 S.: Ill., graph. Darst. Beil.: 1 CD-ROM (12 cm) ISBN: 978-3-642-01621-9.
- [115] Inklinationssensor „3DM“ der Fa. MicroStrain, Inc., Burlington, VT, USA:
www.microstrain.com/3DM.html

- [116] Loffeld, Otmar: Estimationstheorie I / von Otmar Loffeld. - München [u.a.]: Oldenbourg. - (Einführung in die Nachrichtentechnik). - Zugl. Siegen, Univ., Habil.-Schr., 1989 u.d.T.: Loffeld, Otmar: Grundlagen, Konzepte und Anwendungen der Estimationstheorie 1 Grundlagen und stochastische Konzepte, 1990. - XIV, 378, 11 S.: graph. Darst. ISBN: 3-486-21616-3.
- [117] Loffeld, Otmar: Estimationstheorie II / von Otmar Loffeld. - München [u.a.]: Oldenbourg. - (Einführung in die Nachrichtentechnik). - Zugl. Siegen, Univ., Habil.-Schr., 1989 u.d.T.: Loffeld, Otmar: Grundlagen, Konzepte und Anwendungen der Estimationstheorie 2 Anwendungen - Kalman-Filter, 1990. - XIV S., S. 379 - 662: graph. Darst. ISBN: 3-486-21627-9.
- [118] Wenzel, L: Kalman-Filter, Teil 1. In: Elektronik 2000, ISSN 0013-5658, H. 6, S. 64-75.
- [119] Wenzel, L: Kalman-Filter, Teil 2. In: Elektronik 2000, ISSN 0013-5658, H. 8, S. 50-55.
- [120] Wenzel, L: Kalman-Filter, Teil 3. In: Elektronik 2000, ISSN 0013-5658, H. 11, S. 52-58.
- [121] Wenzel, L: Kalman-Filter, Teil 4. In: Elektronik 2000, ISSN 0013-5658, H. 13, S. 74-78.
- [122] C. Netramai, O. Melnychuk, J. Chanin, H. Roth.: Combining PMD a. Stereo camera for Motion Estimation of a Mobile Robot. In The 17th IFAC World Congr., Seoul, Korea, July, 2008.
- [123] Joochim, Chanin: Autonomous navigation and mapping of mobile robots based on 2D/3D cameras combination. - Zsfassung. in dt. Sprache. - Siegen, Univ., Diss., 2011.
- [124] Ni, Melvin Sze-Ming: Three-Dimensional Mapping and Navigation with Stereo Vision. 94 leaves, bound, Thesis (Engineer), Stanford University, 2000.
- [125] Getriebegleichstrommotor „GM9236“ der Fa. Pittman, Harleysville Pennsylvania, USA: www.pittmannet.com
- [126] Mikrocontroller 80C167 der Fa. Infineon: www.infineon.de
- [127] Mikrocontroller 80C167 vertrieben durch die Fa. Phytect: www.phytect.de
- [128] Motor-Control-Boards „MD03“ der Fa. Devantech Ltd.: www.devantech.co.uk
- [129] Motor-Control-Boards „MD22“ der Fa. Devantech Ltd.: www.devantech.co.uk
- [130] Lutz, Holger: Taschenbuch der Regelungstechnik. - 8., erg. Aufl. - Frankfurt am Main: Deutsch, 2010. - 1409 S.: Ill., graph. Darst. - ISBN 978-3-8171-1859-5.
- [131] Lutz, Holger: Taschenbuch der Regelungstechnik: mit MATLAB und Simulink / Holger Lutz; Wolfgang Wendt. - 7., erg. Aufl. - Frankfurt am Main: Deutsch, 2007. - 1356 S.: zahlr. graph. Darst., ISBN: 978-3-8171-1807-6, ISBN: 3-8171-1807-4.
- [132] Zacher, Serge: Automatisierungstechnik kompakt: theoretische Grundlagen, Entwurfsmethoden, Anwendungen. Karl Auer ... Serge Zacher (Hrsg.). - Braunschweig [u.a.]: Vieweg - XV, 501 S.: graph. Darst. - ISBN: 3-528-03897-7, 2000.
- [133] Software Simulink, Zusatz-Software zu Matlab, dient zur Modellierung von technisch, physikalisch, finanzmathematisch, usw. Systemen, Fa. Mathworks: www.mathworks.com
- [134] Echtzeitbetriebssystem VxWorks der Fa. Wind River Systems, Alameda, Kalifornien, USA: www.windriver.com/de
- [135] ICE: C-Bibliothek zur Bildanalyse und Robot Vision: www.inf-cv.uni-jena.de/ice/ice.html
- [136] Schwenkbare USB-Web-Kamera der Fa. Axis: www.axis.com

- [137] Hecht, Eugene: Optik / von Eugene Hecht . - 5., verb. Aufl. . - München : Oldenbourg , 2009 . - XVII, 1125 S. : zahlr. Ill. und graph. Darst. EST: Optics <dt.> . - Aus dem Engl. übers.. - Online-Ausg.: Hecht, Eugene: Optik, ISBN: 978-3-486-58861-3, 2009. (s.a. de.wikipedia.org/wiki/Omnidirektionale_Kamera)
- [138] Rech, Jörg: Wireless LANs : 802.11-WLAN-Technologie und praktische Umsetzung im Detail ; 802.11a/h, 802.11b, 802.11g, 802.11i, 802.11n, 802.11d, 802.11e, 802.11f, 802.11s / Jörg Rech . - 3., aktualisierte und erw. Aufl. . - Hannover : Heise , 2008 . - XX, 572 S. : Ill., graph. Darst. ISBN: 978-3-936931-51-8, 2008. (s.a. de.wikipedia.org/wiki/IEEE_802.11)
- [139] Schnabel, Patrick: Netzwerktechnik-Fibel. Grundlagen, Übertragungstechnik und Protokolle, Anwendungen und Dienste, Sicherheit. 1. Aufl., Norderstedt, Books on Demand, 2004, 308 S.: graph. Darst., ISBN 3-8334-1681-5, www.elektronik-kompodium.de
- [140] Max Riegel, Aik Chindapol, Dirk Kroeselberg: Deploying Mobile WiMAX, John Wiley & Sons, 2009, ISBN 978-0-470-69476-3. (s.a. de.wikipedia.org/wiki/IEEE_802.16)
- [141] Test: Internet-Tempo per UMTS. In: Computer-Bild, 10/2010, Springer, ISSN: 1611-8081.
- [142] Martin Sauter: Grundkurs Mobile Kommunikationssysteme : UMTS, HSDPA und LTE, GSM, GPRS und Wireless LAN / von Martin Sauter . - 4., überarbeitete und erweiterte Auflage . - Wiesbaden : Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH, Wiesbaden, ISBN: 9783834898289, 2011. (s.a. de.wikipedia.org/wiki/Daten%C3%BCbertragungsrate)
- [143] Jörg Stroisch, Thorsten Olscha: Webgrafik-Optimierung. Markt und Technik, München, ISBN 3-8272-6530-4, 2003. (s.a. de.wikipedia.org/wiki/GIF)
- [144] Nelson, Mark: The data compression book / Mark Nelson ; Jean-Loup Gailly . - 2. ed. . - New York, NY : M&T Books. - XIII, 557 S. : Ill, graph. Darst., ISBN: 1-55851-434-1, 1996. (s.a. de.wikipedia.org/wiki/Lempel-Ziv-Welch-Algorithmus)
- [145] Werner, M.: Intersatellite Link (ISL) Networks: Topological Design, Routing and Network Dimensioning. In: Tagungsband. ATM-Sat Workshop, Oberpfaffenhofen, 12. Dez. 2000.
- [146] Satellitentelefonie Iridium: www.iridium.com
- [147] Satellitentelefonie Inmarsat: www.inmarsat.com, broadband.inmarsat.com
- [148] Satellitentelefonie Globalstar: www.globalstar.com
- [149] Satellitentelefonie Thuraya: www.thuraya.com
- [150] Satellitentelefonie Ellipso: www.ellipso.com
- [151] Satellitentelefonie ACeS: www.acesinternational.com
- [152] Satellitentelefonie Orbcomm: www.orbcomm.com
- [153] Satellitentelefonmodem „Explorer 325“ der Fa. Thrane & Thrane: www.thrane.com
- [154] Hans Dodel, Sabrina Eberle: Satellitenkommunikation / by Hans Dodel, Sabrina Eberle . - 2 . - Berlin, Heidelberg : Springer-Verlag, ISBN: 9783540295761 , 2007. (s.a. [de.wikipedia.org/wiki/Iridium_\(Kommunikationssystem\)](http://de.wikipedia.org/wiki/Iridium_(Kommunikationssystem)))
- [155] Mikroprozessoren der Fa. Intel: www.intel.com
- [156] Mikroprozessor „Atom Z530“ der Fa. Intel: www.intel.com/cd/products/services/emea/deu/centrino/atom/387494.htm (s.a. de.wikipedia.org/wiki/Intel_Atom)
- [157] Energieverwaltung in Computern mit ACPI: <http://acpi.info>

- [158] Betriebsmodi S1 - S5 des ACPI-Standards: www.acpi.info/DOWNLOADS/ACPIspec40a.pdf (s.a. de.wikipedia.org/wiki/Advanced_Configuration_and_Power_Interface)
- [159] Wake on LAN (WOL): support.amd.com/us/Embedded_TechDocs/20213.pdf (s.a. de.wikipedia.org/wiki/Wake_On_LAN)
- [160] Mishra, N., K. Chebrolu, B. Raman und A. Pathak: Wake on WLAN. In: WWW '06: Proceedings of the 15th international conference on World Wide Web, S. 761–769, New York, NY, USA, 2006. ACM.
- [161] Shih, E., P. Bahl und M. J. Sinclair: Wake on wireless: an event driven energy saving strategy for battery operated devices. In: MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing a. networking, S. 160–171, New York, USA, 2002. ACM.
- [162] „Wake on Wireless“ und „Wake on Theft“ sind eingetragene Marken der Fa. Atheros Communications, Inc.: www.atheros.com
- [163] WLAN-Chip „AR9280“ mit „Wake on Wireless“-Funktion der Fa. Atheros Communications, Inc.: www.atheros.com
- [164] Mobilfunkmodem-Modul „F5521gw“ mit „Wake on Wireless“-Funktion der Fa. Ericsson: www.ericsson.com
- [165] Tanenbaum, Andrew S.: Computernetzwerke. - 4., überarb. Aufl., [Nachdr.], Verl. Pearson Studium München [u.a.], 949 S. : Ill., graph. Darst., aus dem Engl. übers., ISBN 978-3-8273-7046-4, 2011. (s.a. de.wikipedia.org/wiki/Internetprotokollfamilie)
- [166] Hyper-Terminal-Verbindungen: technet.microsoft.com/en-us/library/bb457166.aspx (s.a. de.wikipedia.org/wiki/HyperTerminal)
- [167] Remotedesktop-Verbindungen: msdn.microsoft.com/de-de/library/aa383015.aspx (s.a. de.wikipedia.org/wiki/Remotedesktopverbindung)
- [168] Web-, WWW- oder http-Server-Software der Apache Software Foundation: www.apache.org
- [169] Web-, WWW- oder http-Verbindungen: www.selfhtml.org
- [170] Web-, WWW- oder http-Verbindungen: www.selfphp.de
- [171] Otto J. Rösch, Klaus Schilling, Hubert Roth: Haptic Interface for Remote Control of Mobile Robots. In: Proceedings 1st IFAC Conference Telematics Applications in Automation and Robotics, FH Ravensburg-Weingarten Juli 2001, Seite 143-147.
- [172] Roth, H., K. Schilling, O. Roesch: Haptic Interfaces for Remote Control of Mobile Robots. In: 15th IFAC World Congress b'02, July 2002, Barcelona Spain, page 212.
- [173] Roth, H., K. Schilling, O. Roesch: Control of Mobile Robots with Force Feedback Devices. In: Mechatronics 2002, June 2002, Enschede Netherlands, ISBN 90 365 17672, page 100.
- [174] Rösch, O.J.; Schilling, K.; Roth, H.: Haptic interfaces fort he remote control of mobile robots. In: Control Engineering Practice, November 2002, in Volume 10, Issue 11, pp. 1309-1313.
- [175] Klaus Schilling, Hubert Roth und Otto Rösch: Fernsteuerung und Telesensorik für mobile Roboter in Raumfahrt, Industrie und Ausbildung. In: at – Automatisierungstechnik 49 (2001) 8, Oldenbourg Verlag, Seite 366-372.
- [176] Klaus Schilling, Hubert Roth and Robert Lieb: Remote Control of a „Mars Rover“ via Internet – to Support Education in Control and Teleoperations. In: Acta Astronautica Vol. 50, No. 3, pp. 173-178. 2002.

- [177] Ruangpayoongsak, Niramon: Development of autonomous features and indoor localization techniques for car-like mobile robots / von Niramon Ruangpayoongsak, 2006. - X, 116 Bl.: Ill., graph. Darst. - Zsfassung in dt. Sprache. - Siegen, Univ., Diss., 2006.
- [178] Hubert Roth, Otto Roesch, Alexander Prusak, Joerg Kuhle: Experiments in mechatronics for remote laboratories. 21st ICDE World Conference on Open Learning & Distance Education, Lifelong Learning in the Networked World; February 2004, Hong Kong, Page 82.
- [179] H. Roth, O. Rösch, A. Prusak: Internetbasierte Laborexperimente für die regelungstechnische Ausbildung. at-Automatisierungstechnik (2003) 11, Oldenbourg Verl., S. 518 - 524.
- [180] H. Roth, A. Prusak, O. Rösch: Internetbasiertes Buchungs- und Verwaltungssystem „Learn2Control“ für Tele-Experimente. In: Telematik 2003. Tagung Siegen, 17. und 18. Juni 2003. VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (Hrsg.), Düsseldorf: VDI-Verlag, 2003 (VDI-Berichte Bd. 1785), ISBN 3-18-091785-7, S. 105 - 114.
- [181] H. Roth, O. Roesch, A. Prusak: Virtual Laboratories for Experiments in Mechatronics. GMA Kongress 2003 „Automation und Information in Wirtschaft und Gesellschaft“, Juni 2003, Baden Baden, Germany, ISBN 3-18-091756-3, S. 331 - 338.
- [182] H. Roth, O. Roesch, A. Prusak, J. Kuhle, A. Hernán González: Mechatronic Experiments in the Engineering Education. (14th EAEEIE International Conference on Innovations in Education for Electrical and Information Engineering), June 2003, Gdansk, Poland, ISBN 83-918622-0-8, page 15.
- [183] Roth, Hubert; Rösch, Otto; Kuhle, Jörg; Prusak, Alexander: Virtual laboratories for control engineering education. (UNESCO (UICEE) Annual Conference on Engineering Education Cairns/Australien February) 2003.
- [184] Roth, Hubert; Rösch, Otto; Kuhle, Jörg; Prusak, Alexander; A. Herán González: Virtuelle Labors for Experimente in der Mechatronik. (UNESCO (UICEE) World Transactions on Engineering and Technology Education Melbourne/Australia) 2002.
- [185] Roth, Hubert; Prusak, Alexander: E-Learning-Portal „Learn2Control“. In: Buletinul Stiintific al Universitatii „Politehnica“ din Timisoara, Romania, Seria Automatica si Calculatoare, Periodica Politehnica, Transactions on Automatic Control and Computer Science 47 (61) (2002).
- [186] H. Roth, O. Roesch, J. Kuhle, A. Prusak, A. Hernán González, G. Georgiev, G. Lehov, S. Stefanova: Remote laboratories for experiments in mechatronics. Monash Engineering Education Series, 6th Baltic Region Seminar on Engineering Education, Sept. 2002, p. 75 – 78.
- [187] K. Schilling, H. Roth, O. Rösch: Mobile Mini-Robots for Engineering Education. Global Journal of Engineering Education. Vol. 6, No. 1, pp. 79-84, Published in Australia.
- [188] H. Roth, O. Roesch, A. Prusak: Virtual Laboratories for Experiments in Mechatronics. GMA Kongress 2003 „Automation und Information in Wirtschaft und Gesellschaft“, Juni 2003, Baden Baden, Germany, ISBN 3-18-091756-3, p. 331 - 338.
- [189] Dillmann, Rüdiger: Autonome mobile Systeme 2009: 21. Fachgespräch, Karlsruhe, 3./4. Dezember 2009 / Rüdiger Dillmann ... (Hrsg.). - Berlin [u.a.]: Springer, 2009. - X, 274 S.: Ill., graph. Darst. - (Informatik aktuell). - Beitr. teilw. dt., teilw. engl., ISBN: 978-3-642-10283-7.
- [190] Berns, Karsten: Autonome mobile Systeme 2007: 20. Fachgespräch, Kaiserslautern, 18./19. Oktober 2007 / Karsten Berns ... (Hrsg.). - Berlin [u.a.]: Springer, 2007. - XII, 348 S.: Ill., graph. Darst. - (Informatik aktuell). - Beitr. teilw. dt., teilw. engl., ISBN: 978-3-540-74763-5.
- [191] Levi, Paul: Autonome Mobile Systeme 2005: 19. Fachgespräch Karlsruhe, 8./9. Dezember 2005 / Paul Levi ... (Hrsg.). - Berlin [u.a.]: Springer, 2006. - X, 332 S.: Ill., graph. Darst. - (Informatik aktuell). - Beitr. teilw. dt., teilw. engl., ISBN: 978-3-540-30291-9 .

- [192] Dillmann, Rüdiger: Autonome Mobile Systeme 2003: 18. Fachgespräch Karlsruhe, 4./5. Dezember 2003 / Rüdiger Dillmann ... (Hrsg.). - Berlin [u.a.]: Springer, 2003. - XI, 343 S.: Ill., graph. Darst. - (Informatik aktuell). - Beitr. teilw. dt., teilw. engl., ISBN: 3-540-20142-4.
- [193] Böge, Alfred: Technische Mechanik: Statik – Dynamik – Fluidmechanik – Festigkeitslehre / von Alfred Böge . - 29., überarbeitete und erweiterte Auflage . - Wiesbaden : Vieweg + Teubner Verlag / Springer Fachmedien Wiesbaden GmbH, Wiesbaden, 2011, ISBN: 9783834881076. (s.a. de.wikipedia.org/wiki/Geometrischer_Schwerpunkt)
- [194] Weber, Jürgen: Ein visuell unterstütztes, laseroptisches Multisensorsystem zur automatisierten Erfassung dreidimensionaler Objekte. Aachen: Shaker, 1998. - IX, 163 S. - (ZESS-Forschungsberichte; Nr. 13). - Zugl.: Siegen, Univ., Diss., 1998, ISBN: 3-8265-4349-1.
- [195] J. Wahrburg, C. Fedrowitz: Robotik. Skript zur Vorlesung „Robotik“, Universität Siegen, 2000.
- [196] Seeber, Günter: Satellite geodesy / Günter Seeber. - 2., completely rev. and extended ed. Berlin [u.a.] : de Gruyter, 2008, ISBN 978-3-11-020008-9. (s.a. de.wikipedia.org/wiki/Global_Positioning_System)
- [197] Bull, Rolf: GPS-Referenzstationsdienste : GPS-Antennen - Koordinatensysteme und Transformation ; ausgesuchte und teilweise überarbeitete Vorträge des GPS-Seminars 2000 des Arbeitskreises GPS im Bildungswerk des VDV / [Red. Bearb.: Rolf Bull] . - Wiesbaden : Chmielorz, 2001 . - 168 S. : zahlr. Ill., graph. Darst., Kt. . - (VDV-Schriftenreihe ; 19), ISBN: 3-87124-261-6. (s.a. de.wikipedia.org/wiki/Differential_Global_Positioning_System)
- [198] Galileo, das europäische Satellitennavigationssystem: www.esa.int/esaNA/galileo.html (s.a. [de.wikipedia.org/wiki/Galileo_\(Satellitennavigation\)](http://de.wikipedia.org/wiki/Galileo_(Satellitennavigation)))
- [199] Ilja N. Bronstein, K. A. Semendjajew, Gerhard Musiol, Heiner Muehlig: Taschenbuch der Mathematik. - 7., vollst. überarb. und erg. Aufl. [d. Neubearb.] . - Frankfurt am Main : Deutsch, 2008 . - XLI, 1221, S., Beil.: 1 CD-ROM (12 cm), ISBN: 978-3-8171-2017-8.
- [200] Tiede, Manfred: Statistik: Regressions- und Korrelationsanalyse / von Manfred Tiede. - München [u.a.]: Oldenbourg, 1987. - XI, 455 S.: graph. Darst., ISBN: 3-486-20528-5.
- [201] Mohr, Richard: Statistik für Ingenieure und Naturwissenschaftler: Grundlagen und Anwendung statistischer Verfahren / Richard Mohr. - Renningen: Expert-Verl., 2003. - V, 202 S.: graph. Darst. - (Kontakt & Studium ; 557), ISBN: 3-8169-2144-2.
- [202] Papula, L.: Mathematik für Ingenieure und Naturwissenschaftler. – Braunschweig [u.a.]: Vieweg, 1994. 3 Vektoranalysis, Wahrscheinlichkeitsrechnung, mathematische Statistik, Fehler- und Ausgleichsrechnung. - 2., verb. Aufl., 1997. - XX, 832 S., ISBN 3-528-14937-X.
- [203] Jiang, Xiaoyi; Bunke, Horst: Dreidimensionales Computersehen: Gewinnung und Analyse von Tiefenbildern. Springer Verl., 1997. - XIII, 361 S., ISBN: 3-540-60797-8.
- [204] Gutmann, Jens S.: Robuste Navigation autonomer mobiler Systeme. Zugl.: Freiburg (Breisgau), Univ., Diss., 2000, Akademische V.-G. Aka, ISBN: 978-3-89838-241-0.
- [205] A. Knoll: Sensordatenfusion für Anwendungen in der Robotik. Manuskript zum Spezialkurs, KI-Frühjahrsschule der GI, 1994.
- [206] R. Smith, M. Self und P. Cheeseman: Estimating Uncertain Spatial Relationships in Robotics. In: I. J. Cox und G. T. Wilfong (Herausgeber): Autonomous Robot Vehicles, S. 167-193, Springer Verl., 1990.
- [207] Mardia, Kanti V.: Multivariate analysis / K. V. Mardia ; J. T. Kent ; J. M. Bibby. - Reprinted . - Amsterdam [u.a.]: Acad. Press, 2003. - XV, 518 S.: graph. Darst. - (Probability and mathematical statistics), ISBN: 0-12-471252-5.

- [208] Fahrmeir, Ludwig; Brachinger, Hans Wolfgang: Multivariate statistische Verfahren / hrsg. von Ludwig Fahrmeir ... Unter Mitarb. von Wolfgang Brachinger ... - 2., überarb. Aufl. - Berlin [u.a.] : de Gruyter, 1996. - XVI, 902 S.: graph. Darst., ISBN: 3-11-013806-9.
- [209] Hartung, Joachim: Multivariate Statistik: Lehr- und Handbuch der angewandten Statistik / von Joachim Hartung und Bärbel Elpelt. - 7., unveränd. Aufl. - München [u.a.]: Oldenbourg, 2007. - XIV, 815 S.: Ill., graph. Darst., ISBN: 3-486-58234-8, ISBN: 978-3-486-58234-5.
- [210] Stahel, Werner A.: Statistische Datenanalyse : eine Einführung für Naturwissenschaftler / Werner A. Stahel . - 5., überarb. Aufl. . - Wiesbaden : Vieweg, 2008 . - XII, 418 S. : Ill., graph. Darst. . - (Studium) , ISBN: 978-3-8348-0410-5.
- [211] Formelsammlung multivariater statistischer Verfahren:
www.prof-roessler.de/Dateien/Statistik/multivariat.pdf
- [212] Dickhaus, Thorsten: Statistische Verfahren für das Data Mining in einem Industrieprojekt. Interner Bericht FZJ-ZAM-IB-2003-08, Forschungszentrum Jülich GmbH, Zentralinstitut für Angewandte Mathematik, 2003.
- [213] M. Ern , Th. Holm: Mathematik I f r Bauingenieure. Vorlesungsskript, Institut f r Algebra, Zahlentheorie und Diskrete Mathematik, Fakult t f r Mathematik und Physik, Leibniz Universit t Hannover, 2008: www.iazd.uni-hannover.de/~erne/Mathematik1
- [214] Katrin Wedekind: Analyse, Implementierung und Evaluation von Matchingverfahren in der medizinischen Bildverarbeitung. Technical Report No. 79/1995, Deutsches Krebsforschungszentrum Abteilung Medizinische und Biologische Informatik (MBI), 1995.
- [215] Wittenberg, T.; Hastreiter, P.: Bildverarbeitung in der Medizin. 2003, Springer Verl., 2003.
- [216] Goshtasby, Ardeshir: 2-D and 3-D image registration for medical, remote sensing, and industrial applications. Hoboken, NJ [u.a.], Verlag Wiley-Interscience, 2005, XV, 258 S.: Ill., graph. Darst., ISBN 0-471-64954-6.
- [217] K. Lingemann: Schnelles Pose-Tracking auf Laserscan-Daten f r autonome mobile Roboter. Master's thesis, University of Bonn, 2004.
- [218] K. Lingemann, A. N chter, J. Hertzberg, and H. Surmann: High-Speed Laser Localization for Mobile Robots. Journal Robotics and Autonomous Systems, (accepted), 2005.
- [219] B. K. P. Horn: Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A, 4(4):629 – 642, April 1987.
- [220] K. S. Arun, T. S. Huang und S.D. Blostein: Least-Squares Fitting of Two 3-D Point Sets, IEEE Transactions on Pattern Analysis and Machine Intelligence, 9 (5), Sept. 1987.
- [221] P. Besl and N. McKay: A method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2):239 – 256, February 1992.
- [222] Y. Chen and G. Medioni: Object Modelling by Registration of Multiple Range Images. In Proceedings of the IEEE Conference on Robotics and Automation (ICRA '91), pages 2724 – 2729, Sacramento, CA, USA, April 1991.
- [223] A. N chter: Autonome Exploration und Modellierung von 3D-Umgebungen, GMD Report 157. GMD, Sankt Augustin, 2002.
- [224] Z. Zhang: Iterative point matching for registration of free-form curves. Technical Report RR-1658, INRIA-Sophia Antipolis, Valbonne Cedex, France, 1992.
- [225] Moravec, H. P.: Towards Automatic Visual Obstacle Avoidance. In: Proceedings of the 5th International Joint Conference on Artificial Intelligence. 1977, S. 584.

- [226] Gimbal-Lock Effekt: www.hq.nasa.gov/office/pao/History/alsj/gimbals.html
(s.a. de.wikipedia.org/wiki/Gimbal_Lock)
- [227] Erik B. Dam, Martin Koch, and Martin Lillholm: Quaternions, Interpolation and Animation. Technical Report DIKU-TR-98/5, University of Copenhagen, July 1998.
- [228] W. Hamilton: On a new Species of Imaginary Quantities connected with a theory of Quaternions. In Proceedings of the Royal Irish Academy, Dublin, Ireland, November 1843.
- [229] Quaternionen: www.grundstudium.info/animation/node17.php
(s.a. de.wikipedia.org/wiki/Quaternion)
- [230] Mobiler Roboter KURT (Kanal-Untersuchungs-Roboter-Testplattform) der Fa. KTO (Kommunikation und Technologietransfer Odenthal): www.kto-web.de
- [231] Herter, Eberhard: Nachrichtentechnik: Übertragung - Vermittlung - Verarbeitung. / Eberhard Herter ; Wolfgang Lörcher. - 9., bearb. Aufl. - München [u.a.]: Hanser, 2004. - 532 S.: zahlr. graph. Darst., ISBN: 3-446-22684-2.
- [232] RoboCup, Robot World Cup Initiative: www.robocup.org
- [233] M. Wiedemann, M. Sauer, F. Driewer, K. Schilling: Analysis and characterization of the PMD camera for application in mobile robotics. In: Proceedings of the 17th World Congress „The International Federation of Automatic Control” Seoul, Korea, July 6-11, 2008.
- [234] M. Goodrich, D. Olsen, J. Crandall, and T. Palmer. Experiments in adjustable autonomy. In: Proceedings of IJCAI Workshop on Autonomy, Delegation and Control: Interacting with Intelligent Agents, 2001.
- [235] C. Nielsen, D. Bruemmer, D. Few, and M. Walton. Mixed-initiative interactions for mobile robot search. In American Association for Artificial Intelligence Mobile Robot Workshop, Boston, Massachusetts, July 2006.
- [236] M. Frank, S. Busch, P. Dietz, and K. Schilling. Teleoperations of a mobile outdoor robot with adjustable autonomy. In Proceedings 2nd ACIDCA-ICMI'2005, International Conference on Machine Intelligence, Tozeur (Tunisia), 2005.
- [237] Sebastian Thrun, Wolfram Burgard, Dieter Fox: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The Mit Press, 668 S., ISBN-10: 0262201623, 2005.
- [238] Andreas Nüchter: 3D Robotic Mapping. Springer-Verlag GmbH, Berlin 2009, ISBN 978-3540898832 (Springer Tracts in Advanced Robotics)).
- [239] Pfad- / Bahnplanungs-Software „MobilePlanner“ von der Fa. Adept, www.adept.de
- [240] Segway Human Transporter der Fa. Segway LLC, Bedford, New Hampshire, USA: www.segway.com
- [241] Microsoft Robotics Developer Studio (MRDS): <http://msdn.microsoft.com/de-de/robotics>
- [242] Handhabungsroboterarm der Fa. Kuka: www.kuka-robotics.de
- [243] Handhabungsroboterarm PA-10 der Fa. Mitsubishi Heavy Industries: www.mhi.co.jp
- [244] Kosko, Bart: Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall, ISBN-10: 0136114350, ISBN-13: 978-0136114352, 1991.
- [245] P. Bahl V. N. Padmanabhan, RADAR: An In-Building RF-Based User Location and Tracking System, Proceedings of IEEE INFOCOM 2000, Vol. 2, Tel-Aviv, Israel (March 2000): S. 775-784.

- [246] Selbstlokalisierung per WLAN mit Fraunhofer Navigator:
www.iis.fraunhofer.de/bf/In/technologie/rssi
bzw. Awiloc: www.awiloc.de
- [247] Selbstlokalisierung per WLAN mit Magicmap, www.magicmap.de
- [248] Selbstlokalisierung per WLAN mit OpenWLANMap, www.openwlanmap.org
- [249] Sebastian Thrun, Computer Science Department, Stanford University:
<http://robots.stanford.edu>
- [250] CeWin, Echtzeiterweiterung für Windows XP: www.kuka-rtos.de
- [251] Kithara RealTime Suite, Echtzeiterweiterung für Windows XP: www.kithara.de
- [252] Real-Time-Linux bzw. Echtzeit-Linux mit Realtime-Preempt-Patches:
Open Source Automation Development Lab (OSADL) eG, www.osadl.org
- [253] Michael Menz: RFID-basierte Positionsbestimmung. Humboldt Universität zu Berlin,
Studienarbeit 2005.
- [254] Drahtlose Alternative in KNX. In: KNX Journal, 1 - 2006, www.konnex.org/news/journal