



TECHNISCHE UNIVERSITÄT CHEMNITZ

---

Fakultät für Elektrotechnik und Informationstechnik

Professur für Nachrichtentechnik

# Diplomarbeit

Robuste Lokalisierung von autonomen Fahrzeugen mittels  
Landmarken

Sebastian Grünwedel

Chemnitz, den 7. Februar 2008

**Betreuer:** Dr.-Ing. Ullrich Scheunert  
Dipl.-Ing. Eric Richter

**Professor:** Prof. Dr.-Ing. Gerd Wanielik

# Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendete Textauschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Chemnitz, den 7. Februar 2008

---

Sebastian Grünwedel

## **Zusammenfassung**

Die Fahrzeuglokalisierung ist im Bereich der Fahrerassistenzsysteme von entscheidender Bedeutung und Voraussetzung für verschiedene Anwendungen der Robotik, wie z.B. Navigation oder Kollisionsvermeidung für fahrerlose Transportsysteme (FTS).

In dieser Arbeit wird ein Verfahren zur Lokalisierung mittels Landmarken vorgestellt, die eine Orientierung bezüglich einer Karte ermöglichen. Dabei werden der Erweiterte-Kalman-Filter und der Partikel-Filter für diese Aufgabe untersucht und verglichen. Ein Schwerpunkt dieser Betrachtungen stellt dabei der Partikel-Filter dar. Die besondere Problematik der Initialisierung wird ausführlich für beide Filter dargestellt.

Simulationen und Versuche zeigen, dass sich der Partikel-Filter für eine robuste Lokalisierung der Fahrzeugposition verwenden lässt. Im Vergleich dazu kann der Erweiterte-Kalman-Filter nur im begrenzten Maße eingesetzt werden.

## **Abstract**

The localization of vehicles is of vital importance in the field of driver assistance systems and a requirement of different applications for robotics, i.e. navigation or collision avoidance for automatic guided vehicle systems.

In this thesis an approach for localization by means of landmarks is introduced, which enables an orientation regarding a map. The extended Kalman filter and the particle filter are analyzed and compared. The main focus for this consideration is on the particle filter. The problematic for initialization is discussed in detail for both filters.

Simulations and tests prove that the particle filter is suitable for robust localization of the vehicle position. Compared to this, the extended Kalman filter can only be used to a certain extend.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>iii</b>
<b>Tabellenverzeichnis</b>	<b>v</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen</b>	<b>4</b>
2.1 Sensoren . . . . .	4
2.1.1 Videokamera . . . . .	4
2.1.2 Inkrementalgeber . . . . .	6
2.1.3 Drehratensensor . . . . .	7
2.2 Landmarken . . . . .	8
2.3 Mathematische Grundlagen . . . . .	9
2.3.1 Affine Transformation . . . . .	10
2.3.2 Kameramodell und -kalibrierung . . . . .	14
2.3.3 Grundlagen der Wahrscheinlichkeitstheorie . . . . .	19
<b>3 Filter &amp; Modelle</b>	<b>26</b>
3.1 Bayes-Filter . . . . .	28
3.1.1 Mathematische Betrachtungen des Bayes-Filters . . . . .	30
3.2 Kalman-Filter . . . . .	31
3.2.1 Kalman-Filter für nichtlineare Systeme . . . . .	34
3.2.2 Systemmodell . . . . .	36
3.2.3 Sensormodell . . . . .	39
3.2.4 Zuordnungsmethoden . . . . .	43
3.2.5 Erläuterungen zur Messwahrscheinlichkeit . . . . .	47
3.3 Partikel-Filter . . . . .	51
3.3.1 Systemmodell & Sensormodell . . . . .	54
3.3.2 Wichtungsfaktor und Normalisierung . . . . .	55
3.3.3 Mathematische Betrachtungen des Partikel-Filters . . . . .	58
3.3.4 Resampling . . . . .	59
3.3.5 Bestimmung des Schätzergebnisses . . . . .	62
3.4 Vergleich Partikel-Filter und Kalman-Filter . . . . .	62
<b>4 Ergebnisse</b>	<b>65</b>
4.1 Initialisierung . . . . .	67

4.1.1	Einfluss der Messkovarianzen . . . . .	70
4.2	Bewegungsmodell . . . . .	73
4.2.1	Diagonalfahrt . . . . .	73
4.2.2	Kurvenfahrt . . . . .	76
4.2.3	Einfluss der Messkovarianzen . . . . .	78
4.3	Testszenario . . . . .	79
<b>5</b>	<b>Software</b>	<b>85</b>
5.1	Struktur der Applikation . . . . .	85
<b>6</b>	<b>Zusammenfassung</b>	<b>90</b>
6.1	Ergebnisse . . . . .	90
6.2	Ausblick . . . . .	91
	<b>Literaturverzeichnis</b>	<b>92</b>

# Abbildungsverzeichnis

1.1	Lokalisierung mit Hilfe eines FTS . . . . .	2
1.2	FTS der Firma FusionSystems GmbH . . . . .	3
2.1	AVT Guppy F-033B/C . . . . .	5
2.2	Funktionsweise 1CCD-Kamera . . . . .	5
2.3	Sensorempfindlichkeit und Bayer-Pattern der Kamera . . . . .	6
2.4	Funktionsweise des Inkrementalgebers . . . . .	7
2.5	Drehratensensor der Firma Melexis . . . . .	8
2.6	Landmarke der Firma FusionSystems . . . . .	9
2.7	Koordinatentransformation: Translation . . . . .	11
2.8	Koordinatentransformation: Rotation . . . . .	13
2.9	Geometrie des Lochkameramodells . . . . .	15
2.10	Bildebene des Lochkameramodells . . . . .	16
2.11	Beispiel einer Lokalisierung eines Roboters . . . . .	20
2.12	Dichtefunktion und Verteilungsfunktion einer Normalverteilung . . . . .	22
3.1	Veranschaulichung des Kalman-Filters . . . . .	34
3.2	Veranschaulichung des Sensormodells . . . . .	40
3.3	Bipartiter Graph . . . . .	44
3.4	Beispiel: Ungarische Methode . . . . .	46
3.5	Beispiel: Nearest-Neighbor-Methode . . . . .	47
3.6	Repräsentation des Partikel-Filters . . . . .	52
3.7	Veranschaulichung der Wichtungsfaktoren des Partikel-Filters . . . . .	56
3.8	Verteilungsdichtefunktion des Partikel-Filters . . . . .	60
4.1	Beispiel für die Darstellung des Partikel-Filters . . . . .	66
4.2	Initialisierungsmethoden des Partikel-Filters . . . . .	68
4.3	Initialisierung des Partikel-Filters . . . . .	69
4.4	Dichteverteilung mehrerer Maxima des Partikel-Filters . . . . .	70
4.5	Praktische Initialisierung des Partikel-Filters . . . . .	71
4.6	Einfluss der Messkovarianzen bei der Initialisierung des Partikel-Filters	72
4.7	Bewegungsmodelluntersuchung einer Diagonalfahrt des Kalman-Filters	74
4.8	Bewegungsmodelluntersuchung einer Diagonalfahrt des Partikel-Filters	75
4.9	Bewegungsmodelluntersuchung einer Kurvenfahrt des Kalman-Filters	76
4.10	Bewegungsmodelluntersuchung einer Kurvenfahrt des Partikel-Filters	77
4.11	Einfluss der Messkovarianzen auf das Fahrverhalten des Kalman-Filters	78

4.12	Einfluss der Messkovarianzen auf das Fahrverhalten des Partikel-Filters	79
4.13	Simulierte Testfahrt für Kalman- und Partikel-Filter . . . . .	80
4.14	Testfahrt mit dem FTS . . . . .	81
4.15	Reale Testfahrt für Kalman- und Partikel-Filter . . . . .	82
4.16	Partikel-Filter-Verteilung mit nur einer Landmarke . . . . .	83
5.1	Struktur der Applikation . . . . .	86
5.2	Screenshot der Anwendung . . . . .	88
5.3	Screenshot der 2D-Anzeige der Anwendung . . . . .	89



# Tabellenverzeichnis

3.1	Algorithmus des kontinuierlichen Bayes-Filters . . . . .	29
3.2	Algorithmus des finiten Bayes-Filters . . . . .	30
3.3	Algorithmus des Kalman-Filters . . . . .	33
3.4	Algorithmus des Erweiterten-Kalman-Filters . . . . .	36
3.5	Vergleich EKF und Kalman-Filter . . . . .	37
3.6	Algorithmus für unabhängige Messungen . . . . .	49
3.7	Algorithmus für abhängige Messungen . . . . .	51
3.8	Algorithmus des Partikel-Filters . . . . .	53
3.9	Low Variance Resampling Algorithmus . . . . .	60
4.1	Simulierter Fehler der Filter . . . . .	81
4.2	Praktischer Fehler der Filter . . . . .	83

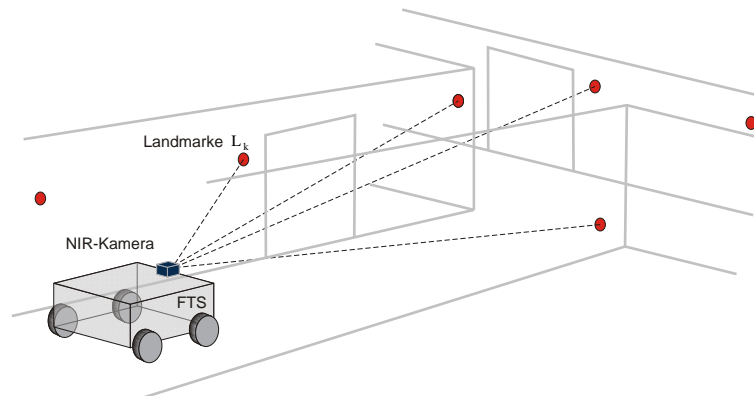
# 1 Einleitung

Im Bereich der Fahrerassistenzsysteme liegt ein besonderer Schwerpunkt auf dem Erfassen des Fahrzeugumfeldes. Aufgrund der Vielfalt unterschiedlichster Sensoren, wie zum Beispiel visuellen Sensoren oder Distanzsensoren und anderen Datenerfassungselementen, werden verschiedenste Daten aufgenommen. Diese Daten werden von unterschiedlichen Applikationen gleichzeitig genutzt bzw. benötigt.

Die Entwicklung im Bereich der Fahrerassistenzsysteme führt zunehmend zu verbesserter Sicherheit und Komfort von Automobilen. Durch den Einsatz entsprechender Systeme lassen sich vermehrt die Fahrunfälle (ohne Einwirkung anderer Verkehrsteilnehmer) verringern. Die Liste der Anwendungen der im Fahrzeug einsetzbaren Systeme ist lang. Systeme wie ABS (Antiblockiersystem) oder ESP (Elektronisches Stabilitätsprogramm) werden heute meist serienmäßig verbaut. Durch die Entwicklung vieler neuer Systeme können der Fahrkomfort sowie kleine körperliche Defizite kompensiert werden und somit auch gehandicapten und älteren Personen mehr Mobilität zu ermöglichen.

Die Fahrzeuglokalisierung ist im Bereich der Fahrerassistenzsysteme von entscheidender Bedeutung. Erst durch Kenntnis der aktuellen Position lassen sich neue Systeme, wie Navigation oder Kollisionsvermeidung, entwickeln. Durch das Zusammenwirken entsprechender Systeme ist auch autonomes Fahren möglich. Die Bestimmung der Eigenposition kann auf verschiedene Arten erfolgen. In einfachen Navigationssystemen wird meist das GPS (Global Positioning System) angewandt, das aber häufig nur eine unzureichende Genauigkeit aufweist (Umkreis von 10 - 30 m). Die Zuverlässigkeit der GPS-basierten Systeme wird durch verschiedene Faktoren, wie stark bebaute Gebiete oder Tunnel, beeinflusst. Eine genaue Positionsbestimmung mit funkbasierten Systemen ist auf Grund der Zuverlässigkeit und Ungenauigkeit für viele Anwendungen, wie Indoor-Navigation, nicht geeignet.

Aus diesem Grund müssen andere Methoden für die Fahrzeuglokalisierung eingesetzt werden. Diese Arbeit beschäftigt sich mit einer Methode der Fahrzeuglokalisierung mittels Landmarken. Mit Hilfe der Landmarken ist die Orientierung bezüglich einer



**Abbildung 1.1:** Lokalisierung mit Hilfe eines FTS der Firma FusionSystems GmbH

Karte möglich (Abbildung 1.1). Für eine erfolgreiche Navigation ist die Kenntnis der Fahrzeugposition und dessen Ausrichtung notwendig. In der Diplomarbeit sollen sowohl der Extended-Kalman-Filter als auch der Partikel-Filter für diese Aufgabe untersucht und verglichen werden. Die entsprechenden Programmibliotheken wurden in der Programmiersprache „C/C++“ entwickelt und die Applikation (GUI) wurde mit dem *.net*-Framework<sup>1</sup> umgesetzt. Die Arbeit gliedert sich in folgende Kapitel:

## 1. EINLEITUNG

In diesem Kapitel wird die Aufgabe der Arbeit beschrieben sowie kurz auf das verwendete fahrerlose Transportsystem (FTS) eingegangen.

## 2. GRUNDLAGEN

Um die Filter erklären zu können, werden in diesem Abschnitt die verwendeten Sensoren und mathematische Grundlagen erläutert.

## 3. FILTER & MODELLE

Dieses Kapitel beschreibt die beiden verwendeten Filter und die zugrundeliegenden Modelle. Besonders wird dabei auf den Partikel-Filter und seine Eigenschaften eingegangen.

## 4. ERGEBNISSE

Die Ergebnisse beider Filter sowie deren Vor- und Nachteile in Bezug auf die gestellte Aufgabe werden in diesem Abschnitt erläutert.

---

<sup>1</sup>*.net* ist eine von Microsoft entwickelte Softwareplattform.

5. SOFTWARE

Das fünfte Kapitel beschäftigt sich mit der softwaretechnischen Umsetzung sowie der eigentlichen Applikation.

6. ZUSAMMENFASSUNG



**Abbildung 1.2:** FTS der Firma FusionSystems GmbH

Das von der Firma FusionSystems GmbH bereitgestellte Testfahrzeug (FTS) (Abbildung 1.2) basiert auf einem Fahrzeug der Götting KG und wurde mit folgenden Sensoren ausgestattet bzw. ergänzt:

- Drehratensensor
- Inkrementalgeber zur Messung der Geschwindigkeit
- Videokamera
- Sick-Laserscanner zur Hinderniserkennung

Mit Hilfe dieser Sensoren ist es möglich das Fahrzeug bezüglich einer Karte zu lokalisieren. Durch eine geschickte Datenfusion wird der Informationsgehalt aller Sensoren verknüpft und optimal ausgewertet.

## 2 Grundlagen

In dem folgenden Kapitel werden die verwendeten Sensoren vorgestellt und wichtige mathematische Grundlagen für die weiteren Kapitel erklärt. Der wichtigste Sensor für diese Aufgabe ist die Videokamera, da dieser Sensor von beiden Filtern (Partikel- und Kalman-Filter) genutzt wird. Die beiden anderen Sensoren (Inkrementalgeber und Drehratensensor) sind für diese bestimmte Aufgabe nur für den Partikel-Filter von Bedeutung.

Die mathematischen Grundlagen beinhalten die Transformation für die Kamera sowie bestimmte Ausgangspunkte, um später die Filter und Filtermodelle verstehen zu können.

### 2.1 Sensoren

Allgemein versteht man unter einem Sensor ein technisches Bauteil, das bestimmte Eigenschaften als Messgröße quantitativ erfasst. Die für diese Aufgabe verwendeten Sensoren besitzen alle unterschiedlichen Informationsgehalt und müssen auf geeignete Weise miteinander fusioniert werden. Dabei nutzt der Partikel-Filter alle drei Sensoren ausgiebig. Der Erweiterte-Kalman-Filter arbeitet hier in dieser Aufgabe nur mit dem Videosensor zur Positionsbestimmung und versucht die Geschwindigkeit und Drehrate selbst zu schätzen. Im Folgenden werden nun die einzelnen Sensoren näher erläutert.

#### 2.1.1 Videokamera

Die Aufnahme der benötigten Bilder erfolgte mit einer CCD-Bayer-Pattern-Kamera (AVT Guppy F-033B/C) der Firma Allied Vision Technology (Abbildung 2.1). Diese Kamera nutzt den IEEE 1394 Standard<sup>1</sup> (FireWire) zur Übertragung der Bildinformationen. Als Schnittstelle für die Übertragung wird der DCAM-Standard<sup>2</sup> verwendet

---

<sup>1</sup>FireWire ist eine von der Firma Apple entwickelte digitale Schnittstelle.

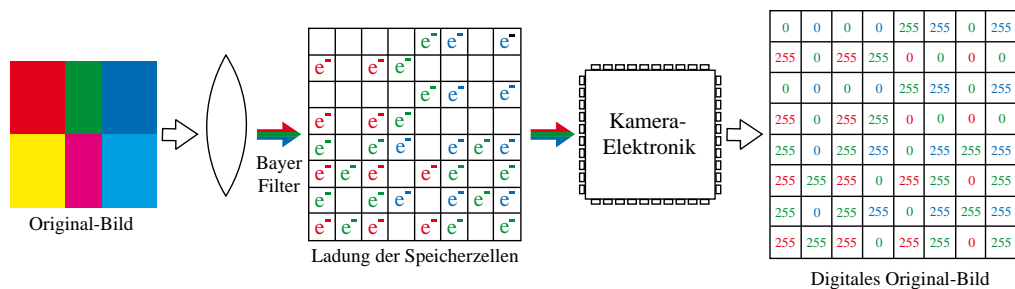
<sup>2</sup>DCAM steht für „1394-based Digital Camera Specification“ und wurde von der 1394 Trade Association definiert [DCAM07].

[DCAM07]. Somit ist es möglich plattformunabhängig die Bilddaten zu einem entsprechenden System zu übertragen.



**Abbildung 2.1:** CCD-Farbkamera (AVT Guppy F-033B/C) der Firma Allied Vision Technology [Guppy07]

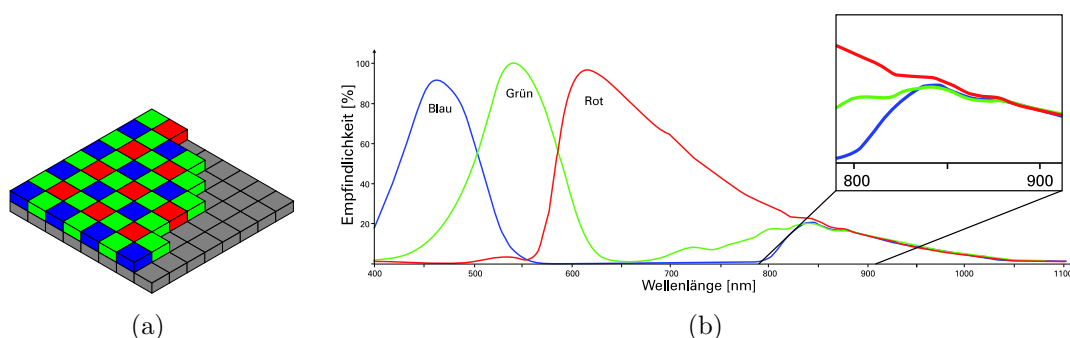
Die Funktionsweise von CCD-Kameras ist einfach. Der Sensor ist eine Art Speicher-Chip. Mit Hilfe des Fotoeffekts werden durch Lichtstrahlen Ladungen in den einzelnen Speicherzellen erzeugt. Diese Ladungen werden nach der Belichtungszeit von der Kameraelektronik ausgelesen und aufbereitet. Die Kameraelektronik erzeugt ein digitales Bild, welches nun zur Verfügung steht und weiter verarbeitet werden kann (Abbildung 2.2) [ImgScr07].



**Abbildung 2.2:** Schematische Darstellung der Funktionsweise einer 1CCD-Kamera [Guppy07]

Um farbige Bilder mit einer CCD-Kamera erhalten zu können, sind drei CCD-Sensoren nötig. Diese drei Sensoren sind empfindlich für den RGB-Farbraum - Rot, Grün und Blau-Komponente. Um Bandbreite für die Übertragung und die Kosten für

die Sensoren zu senken, wurde die 1CCD-Farbkamera entwickelt. Diese Kamera besitzt nur einen Sensor, auf welchen sich mosaikähnlich verschiedene Farbfilter verteilen. Man spricht hier von einem Bayer-Filter (Bayer-Pattern) (Abbildung 2.3(a)). Das Bayer-Filter-Muster ist ein sich wiederholendes  $2 \times 2$ -Mosaik, welches aus Rot-, Grün und Blau-Komponenten besteht. Das bedeutet, dass jede Speicherzelle für nur eine Komponente empfindlich ist [ImgScr07]. Mit Hilfe der Farb-Interpolation werden die entsprechenden Farben für jedes Pixel berechnet. Somit ergibt sich, wie bei einer 3CCD-Kamera, ein Farbbild, welches weiterverarbeitet werden kann.



**Abbildung 2.3:** (a) Bayer-Pattern eines Kamerasensors in der Form RG-GB, (b) Sensorempfindlichkeit für die Rot-, Grün- und Blaukomponente

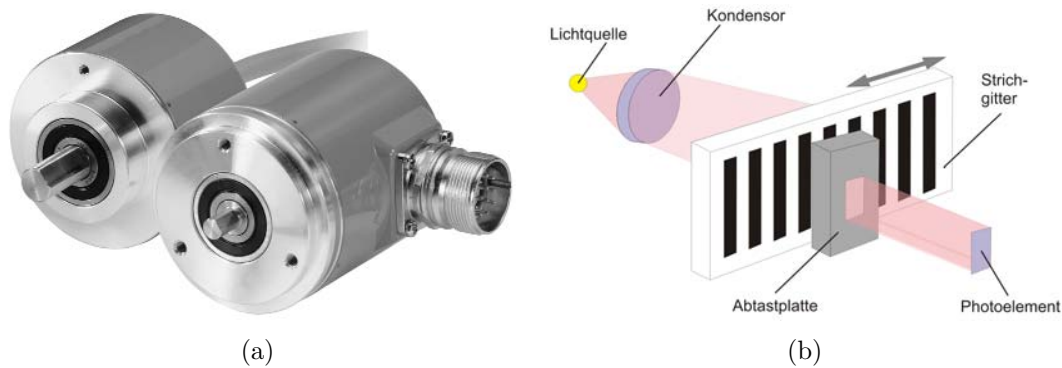
Für die Aufgabe wurde weiterhin ein Infrarotfilter (850nm) verwendet um die Landmarken (siehe Abschnitt 2.2) extrahieren zu können. In der Abbildung 2.3(b) ist dargestellt, dass die Rot-, Grün- und Blausensoren im Bereich des Infrarotfilters (850nm) ungefähr die gleiche Sensitivität aufweisen. Aus diesem Grund lässt sich die Kamera als Grauwertkamera für infrarotes Licht benutzen. Mit Hilfe dieser Informationen lassen sich die Landmarken relativ einfach aus dem Bild extrahieren.

### 2.1.2 Inkrementalgeber

Als Inkrementalgeber wird ein Sensor bezeichnet, der zur Erfassung von Lageänderungen dient. Dieser Sensor ist in der Lage die Wegstrecke als auch die Richtung aufzunehmen. Für die Aufgabe wurde ein Inkrementalgeber der Firma Kübler [Kue07] verwendet.

Mit Hilfe einer rotierenden Scheibe mit lichtdurchlässigen Schlitzen (Inkrementen), einer davor liegenden Blende und einer Fotozelle werden durch eine Folgeelektronik

binäre Ausgangssignale erzeugt. Je nach Typ wird entweder ein Zählvorgang durch die Änderung des Lichtstroms oder eine Erkennung eines binären Musters, die einer definierten Position der Scheibe zugeordnet werden kann, ausgeführt (Abbildung 2.4(b)).



**Abbildung 2.4:** (a) Inkrementalgeber der Firma Kübler, (b) Schematische Darstellung der Funktionsweise eines Inkrementalgebers [Intgeb07]

Durch die Anzahl  $n$  der Inkremente, die über einen Counter gezählt werden können, und dem bekannten Weg  $s_i$  für ein Inkrement lässt sich die Geschwindigkeit  $v$  mit folgender Formel für eine Zeitspanne  $T$  berechnen.

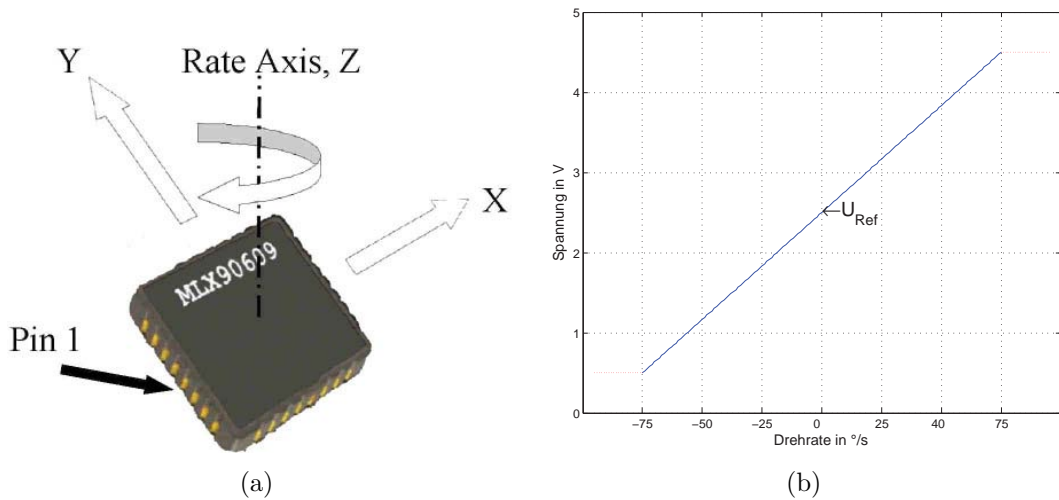
$$v = \frac{n \cdot s_i}{T} \quad (2.1)$$

Bei dieser Formel werden Fehler für Kurvenfahrten nicht berücksichtigt.

### 2.1.3 Drehratensensor

Für das FTS wurde ein Drehratensensor (MLX90609) der Firma Melexis [Mel07] verwendet (Abbildung 2.5(a)). Die Drehrate entspricht dabei der Richtungsänderung pro Zeiteinheit und wird in Grad pro Sekunde gemessen. Der Sensor misst die bei einer Drehung auftretende Corioliskraft und gibt die zur Drehrate direktproportionale Spannung zurück (Abbildung 2.5(b)). Der Messbereich des Sensor liegt bei  $\pm 75^\circ/s$ .





**Abbildung 2.5:** (a) Drehratensensor der Firma Melexis, (b) Abhängigkeit Spannung - Drehwinkel des Drehratensensors [Mel07]

## 2.2 Landmarken

Als Landmarke bezeichnet man ein meist auffälliges sichtbares Objekt. Für die Aufgabe wurde von der Firma FusionSystems GmbH, Chemnitz, eine spezielle Landmarke (Abbildung 2.6(b)) entwickelt, die aus einer LED besteht. Diese LED sendet infrarotes Licht (850 nm) aus und kann somit von der verwendeten Kamera mit Infrarotfilter empfangen werden. Mit Hilfe dieser Landmarken, deren Position und der Erkennungssoftware für die Kamera ist es möglich, das verwendete FTS bezüglich einer Karte zu lokalisieren. Eine solche Landmarke ist durch verschiedene Merkmale charakterisiert (Abbildung 2.6(a)):

- Position  $L_{\text{Pos}}$  bezüglich eines Koordinatensystems
- Richtung  $\underline{r}$  der Ausstrahlung der LED
- horizontaler und vertikaler Abstrahlungswinkel ( $\alpha_v$  und  $\alpha_h$ )

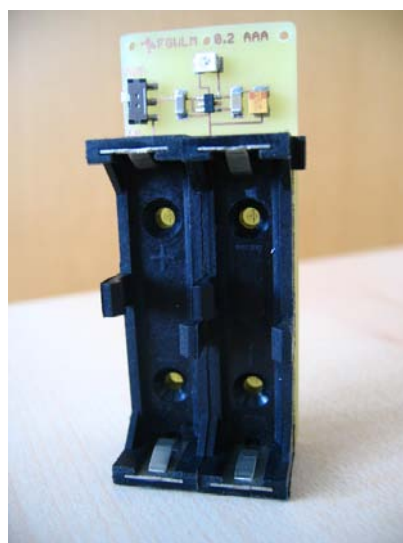
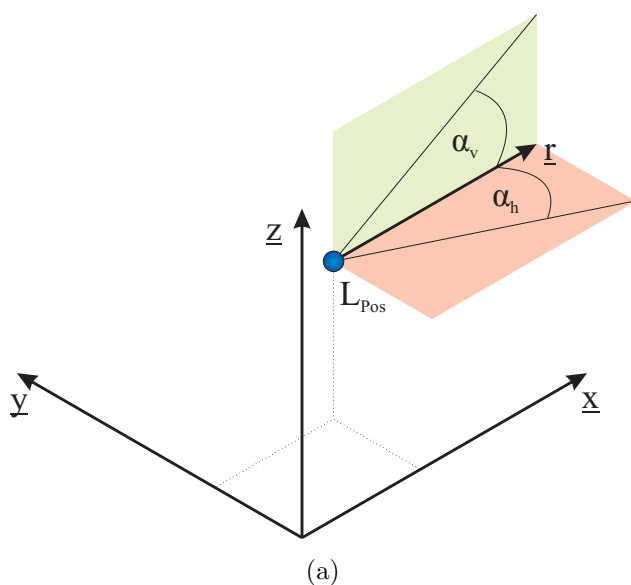
Somit lässt sich eine solche Landmarke wie folgt beschreiben:

$$L = \{L_{\text{Pos}}, \underline{r}, \alpha_v, \alpha_h\} \quad (2.2)$$

Für die Beschreibung einer LED-Landmarke können auch noch weitere Merkmale hinzugenommen werden. So wäre es beispielsweise möglich den Sichtbereich einer

Landmarke noch weiter einzuschränken (Abschirmung der Ausstrahlung einer LED) und diesen über einen gerichteten Lichtstrahl zu beschreiben. In der Aufgabe wurde jedoch die einfachste Beschreibung einer LED-Landmarke, die nur durch ihre Position bezüglich eines Koordinatensystems charakterisiert ist, verwendet. Dabei lässt sich die  $k$ -te Landmarke  $L_k$  durch ihre Koordinaten bezüglich eines Koordinatensystem  $w$  wie folgt beschreiben:

$$L_k = [{}^w x, {}^w y, {}^w z]^T \quad (2.3)$$



**Abbildung 2.6:** Veranschaulichung einer Landmarke: (a) Merkmale einer Landmarke, (b) Landmarke der Firma FusionSystems GmbH, Chemnitz

## 2.3 Mathematische Grundlagen

Für das Verständnis der Filter und Sensormodelle sind einige mathematische Grundlagen notwendig. Es werden im Folgenden die Koordinatentransformation sowie einige wahrscheinlichkeitstheoretische Ansätze, die insbesondere für das Verständnis der Filter erforderlich sind, näher erläutert.

### 2.3.1 Affine Transformation

Mit Hilfe von homogenen Koordinaten ist die Darstellung von affinen Transformationen auf einfache Weise möglich. So lassen sich Operationen linear ausdrücken, indem das Problem in einen höheren Raum transformiert wird.

Ein Punkt  $\underline{x} = (x, y, z)^T$  in einem 3-dimensionalen Raum wird durch homogene Koordinaten als ein 4-dimensionaler Vektor veranschaulicht. Ein homogener Vektor

$$\underline{x} = (x_1, x_2, x_3, x_4)^T$$

mit  $x_4 \neq 0$  repräsentiert einen Punkt  $(x, y, z)^T$  in  $\mathbb{R}^3$  mit inhomogenen Koordinaten:

$$x_{\text{inh}} = \frac{x_1}{x_4}, y_{\text{inh}} = \frac{x_2}{x_4}, z_{\text{inh}} = \frac{x_3}{x_4} \quad (2.4)$$

Die homogene Darstellung eines Punktes  $\underline{x} = (x, y, z)^T$  entspricht:

$$\bar{x} = (x, y, z, 1)^T \quad (2.5)$$

Homogene Punkte mit  $x_4 = 0$  beschreiben Punkte im Unendlichen. Affine Transformationen<sup>3</sup> beinhalten Abbildungen zwischen zwei Vektorräumen, die im Allgemeinen aus einer linearen Transformation  $A$  und einer Translation  $t$  bestehen. Eine Abbildung  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  der Form

$$F(\bar{x}) := A \cdot \underline{x} + \underline{t}$$

mit  $\underline{x} \in \mathbb{R}^n, \underline{t} \in \mathbb{R}^m, A \in \text{Mat}(m, n, \mathbb{R})$  heißt eine affine Abbildung von  $\mathbb{R}^n$  in  $\mathbb{R}^m$ . Dabei wird  $A \cdot \underline{x}$  als der lineare Anteil und  $\underline{t}$  als der Translationsanteil bezeichnet. Unter Verwendung homogener Koordinaten lässt sich diese Gleichung bezogen auf  $\mathbb{R}^3$  durch eine nicht-singuläre  $4 \times 4$ -Matrix folgendermaßen darstellen:

$${}^{K'}\underline{x} = [A|\underline{t}] \cdot {}^K\underline{x} \quad (2.6)$$

Für zwei Koordinatensysteme  $K$  und  $K'$  beschreibt diese Gleichung die Transformation eines Punktes  ${}^K\underline{x} \in K$  in einen neuen Punkt  ${}^{K'}\underline{x} \in K'$ . Für weiterführende Informationen bezüglich affinen Transformationen und Geometrie im Bereich Computer Vision sei auf [Hartley03] verwiesen.

<sup>3</sup>Affine Transformationen sind ein Spezialfall der projektiven Abbildung.

### Translation

Als Translation wird hier die Verschiebung des Koordinatenursprungs durch einen Verschiebungsvektor angesehen. Für die Translationmatrix in homogenen Koordinaten ergibt sich folgende Block-Form:

$${}^K \underline{x} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^{K'} T_K} {}^{K'} \underline{x} \text{ bzw. } {}^{K'} \underline{x} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & -t_1 \\ 0 & 1 & 0 & -t_2 \\ 0 & 0 & 1 & -t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{{}^K T_{K'}} {}^K \underline{x} \quad (2.7)$$

Es werden zwei kartesische Koordinatensysteme  $K$  und  $K'$  betrachtet, wobei  $K'$  um den Vektor  ${}^K \underline{t} = (t_1, t_2, t_3, 1)$  bezüglich  $K$  verschoben ist. Ein beliebiger Punkt  ${}^K \underline{x}$  multipliziert mit der Translationsmatrix 2.7 ergibt dabei den neuen Punkt  ${}^{K'} \underline{x}$  bezüglich  $K'$  (Abbildung 2.7).

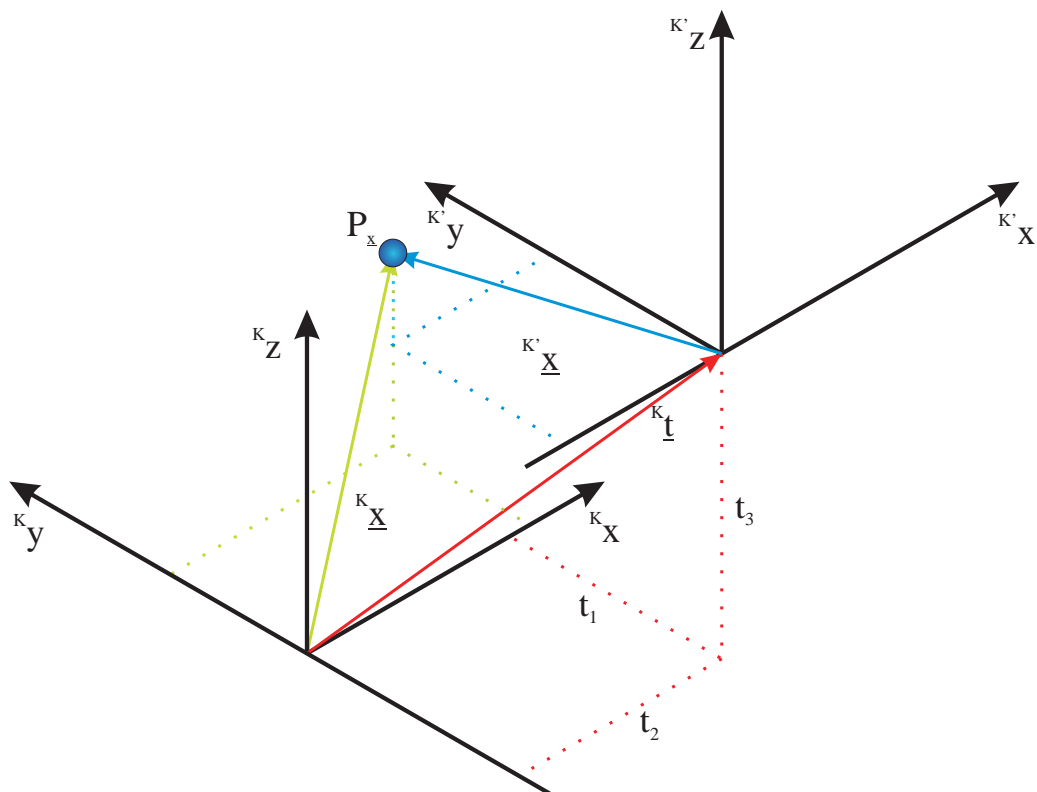


Abbildung 2.7: Translation des Punktes  ${}^K \underline{x}$  zu dem neuen Punkt  ${}^{K'} \underline{x}$

**Rotation**

Bei der Rotation wird das Koordinatensystem um einen Winkel  $\alpha$  einer Achse gedreht. Für die Betrachtung wird ein beliebiger Punkt  ${}^K \underline{x}$  aus dem Ausgangskordinatensystem  $K$  in einen neuen Punkt  ${}^{K'} \underline{x}$  eines gedrehten Koordinatensystems  $K'$  transformiert. Dabei ergeben sich für den realen Raum ( $\mathbb{R}^3$ ) drei Rotationsmatrizen, die der Drehung um die entsprechende Achse entsprechen:

- Rotation um die x-Achse

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

- Rotation um die y-Achse

$$R_y = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

- Rotation um die z-Achse

$$R_z = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

Durch Kombination dieser Rotationsmatrizen lässt sich jede beliebige Drehung im  $\mathbb{R}^3$  durchführen. Dabei ist zu beachten, dass das Kommutativgesetz bei der Kombination der einzelnen Matrizen miteinander nicht gilt. Das bedeutet, es ist ein Unterschied, ob als erstes um die x-Achse und dann die y-Achse rotiert wird oder umgekehrt.

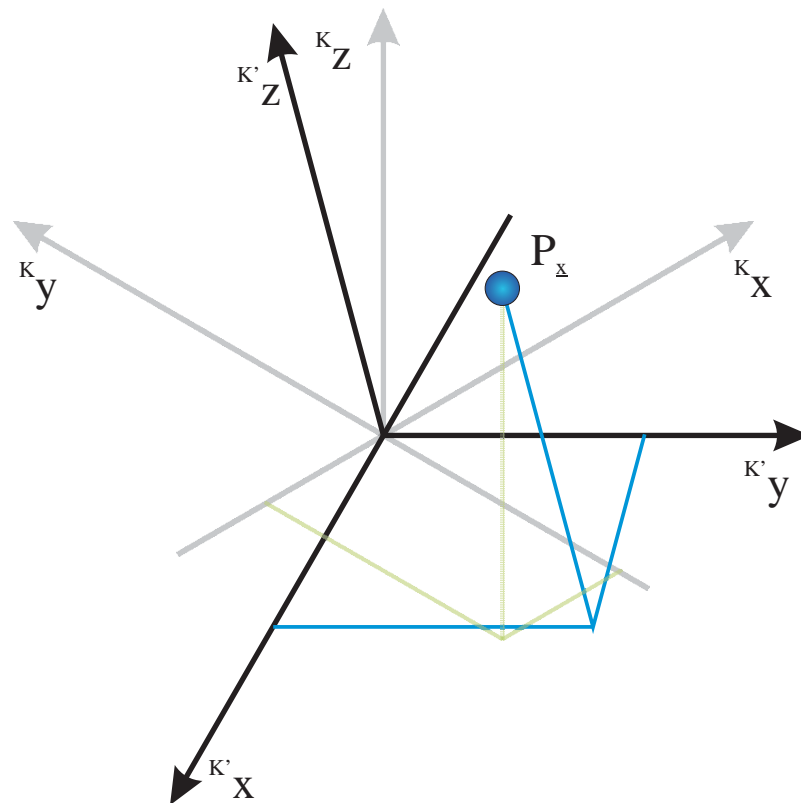


Abbildung 2.8: Rotation des Punktes  ${}^K \underline{x}$  zu dem neuen Punkt  ${}^{K'} \underline{x}$

### Koordinatentransformation

Koordinatentransformationen beschreiben die Beziehungen zwischen Koordinatensystemen. Das bedeutet, ein beliebiger Punkt  ${}^K \underline{x}$  aus dem Koordinatensystem  $K$  kann über eine Transformation mit einem Punkt  ${}^{K'} \underline{x}$  aus einem anderen Koordinatensystem  $K'$  beschrieben werden. Für diese Betrachtungen sind affine Transformationen geeignet. Für eine Koordinatentransformation werden meist eine Translation und verschiedene Rotationen miteinander kombiniert. Die sich daraus ergebende affine Transformation hat folgende Form:

$${}^{K'} \underline{x} = A \cdot {}^K \underline{x} \text{ bzw. } {}^K \underline{x} = A^{-1} \cdot {}^{K'} \underline{x} \quad (2.11)$$

Durch Invertierung der Transformationsmatrix ist eine Rücktransformation möglich. Ein Beispiel für eine Koordinatentransformation wäre eine Verschiebung eines kartesischen Koordinatensystems mit anschließender Rotation um die x- und y-Achse.

Mit Hilfe der Translationsmatrix  $T$  (Gleichung 2.7) und der Rotationsmatrizen  $R_x$  (Gleichung 2.8) und  $R_y$  (Gleichung 2.9) ergibt sich die resultierende Matrixform:

$$\begin{aligned}
 {}^K \underline{x} &= A^{-1} \cdot {}^{K'} \underline{x} \\
 {}^K \underline{x} &= {}^K T_{K'} \cdot R_x \cdot R_y \cdot {}^{K'} \underline{x} \\
 {}^K \underline{x} &= \begin{bmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot {}^{K'} \underline{x}
 \end{aligned}$$

Mit Hilfe solcher affinen Transformationen lassen sich beliebige Punkte von einem Koordinatensystem in ein anderes transformieren.

### 2.3.2 Kameramodell und -kalibrierung

Das Prinzip einer Kamera besteht darin, die 3D-Umgebung auf eine 2D-Bildebene zu projizieren. Für diese Projektion ist ein Modell einer realen Kamera notwendig. In dem folgenden Abschnitt wird sowohl das Kameramodell als auch eine für dieses Modell notwendige Kalibrierung erklärt.

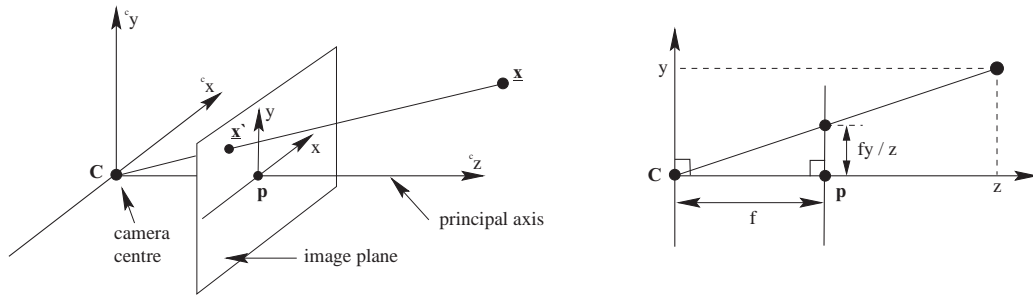
#### Kameramodell

Es gibt verschiedene Kameramodelle. Dieser Abschnitt beschäftigt sich mit dem sogenannten Lochkameramodell, welches für CCD-Sensoren entwickelt wurde. Die in Abschnitt 2.1.1 verwendete CCD-Kamera lässt sich so mit Hilfe des Lochkameramodells<sup>4</sup> beschreiben. Betrachtet wird die Zentralprojektion von 3D-Punkten auf eine Ebene. Dabei ist der Koordinatenursprung der Projektion ein euklidisches Koordinatensystem. Die Ebene mit der Gleichung  $z = f$ , mit  $f$  als Brennweite der Kamera, wird als Bildebene bezeichnet. Über das Lochkameramodell wird ein Punkt im Raum mit den Koordinaten  $\underline{x} = [x, y, z]^T$  auf einen Punkt  $\underline{x}' = [fx/z, fy/z, f]^T$  in der Bildebene abgebildet. Dies geschieht über eine Gerade zwischen dem Ursprung der Projektion und dem 3D-Punkt selbst (Abbildung 2.9).

Auf diese Weise wird ein Punkt aus dem 3-dimensionalen Euklidischen Raum  $\mathbb{R}^3$  in

---

<sup>4</sup>engl. *pinhole model*



**Abbildung 2.9:**  $C$  ist das Kamerazentrum und  $p$  der Bildmittelpunkt. Das Kamerazentrum ist hier auch gleichzeitig der Koordinatenursprung [Hartley03].

den 2-dimensionalen Euklidischen Raum  $\mathbb{R}^2$  abgebildet. Für die Abbildung gilt:

$$[x, y, z]^T \mapsto [fx/z, fy/z, f]^T \quad (2.12)$$

Der Ursprung der Projektion wird als Kamerazentrum (*camera centre*), die Achse, die senkrecht auf der Bildebene (*image plane*) steht, als *principle axis* und der Punkt, an dem diese Achse die Bildebene schneidet, als Bildmittelpunkt (*principle point*) bezeichnet.

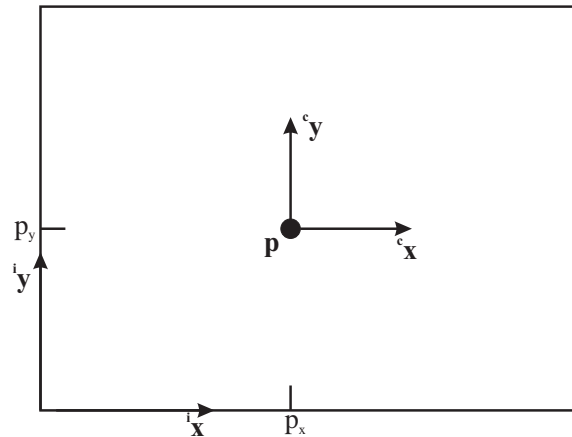
Für die Darstellung einer solchen Projektion in homogenen Koordinaten ergibt sich folgende Formel:

$$\begin{bmatrix} c_x \\ c_y \\ c_z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} i_x \\ i_y \\ c_z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ c_z \\ 1 \end{bmatrix} \quad (2.13)$$

Dabei bezeichnen  $c$  die Koordinaten im 3-dimensionalen Kamerakoordinatensystem und  $i$  die Bildkoordinaten im Bild. Die homogene  $3 \times 4$ -Matrix wird auch häufig als Kameraprojektionsmatrix bezeichnet. Da der Koordinatenursprung der Bildebene nicht im Bildmittelpunkt  $(p_x, p_y)$  liegt, ergibt sich durch Verschiebung des Koordinatensystems folgende Projektionsmatrix (Abbildung 2.10):

$$\begin{bmatrix} i_x \\ i_y \\ c_z \end{bmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ c_z \\ 1 \end{bmatrix} \quad (2.14)$$





**Abbildung 2.10:** Bild-  $(i_x, i_y)$  und Kamerakoordinatensystem  $(c_x, c_y)$  [Hartley03]

Das Lochkameramodell nimmt an, dass die Bildkoordinaten Euklidische Koordinaten darstellen und für beide Achsen gleich skaliert sind. Für CCD-Kameras besteht die Möglichkeit, auch nicht-quadratische Pixel zu verwenden. So ergeben sich Skalierungsfaktoren  $m_x$  und  $m_y$  für jede Richtung, die die Anzahl der Pixel pro Längeneinheit beschreiben. Desweiteren kann es vorkommen, dass die Pixelachsen nicht senkrecht aufeinander stehen. Die Verdrehung dieser Achsen zueinander wird auch als *Skew-Parameter*  $s$  bezeichnet. Im Allgemeinen lassen sich das Lochkameramodell, die Skalierungsfaktoren und der Skew-Parameter als *finite projective Kamera*<sup>5</sup> zusammenfassen:

$$\begin{bmatrix} i_x \\ i_y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ c_z \\ 1 \end{bmatrix}, \quad (2.15)$$

wobei  $\alpha_x = f \cdot m_x$  und  $\alpha_y = f \cdot m_y$  die Brennweite abhängig von den Pixeldimensionen für jede Achse beschreiben. Gleichermäßen stellt  $(x_0, y_0)$  den Bildmittelpunkt abgeleitet aus den Pixeldimensionen  $x_0 = m_x \cdot p_x$  und  $y_0 = m_y \cdot p_y$  dar. Für die Bestimmung der intrinsischen Parameter einer Kamera ist die Kalibrierungsmatrix  $K$  zu bestimmen:

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.16)$$

<sup>5</sup>engl. *finite projective camera*

## Kamerakalibrierung

Als Kamerakalibrierung versteht man die Bestimmung der intrinsischen und extrinsischen Parameter einer Kamera. Diese werden über numerische Verfahren aus einer Menge von korrespondierenden Punkten bestimmt. So lassen sich beispielsweise mit Hilfe eines bekannten Schachbrettmusters Aufnahmen mit einer Kamera durchführen und danach in jedem Bild Beziehungen zwischen dem bekannten Schachbrettmuster und den dazugehörigen Bildpunkten herstellen. Auf diese Weise lassen sich die nötigen Parameter einer Kamera bestimmen. Als intrinsische Parameter werden die Parameter der Kalibrierungsmatrix einer Kamera bezeichnet. Diese sind für ein bestehendes Kamera-Linsen-System konstant. Als extrinsische Parameter wird die Projektionsmatrix bezeichnet, die einen 3D-Weltpunkt in das Kamerakoordinatensystem überführt. Diese Parameter sind von Frame zu Frame unterschiedlich, da diese von vielen äußeren Bedingungen, wie z.B. Bewegung der Kamera, abhängen.

Für nähere Informationen zur numerischen Berechnung der Parameter sowie weiterführende Erläuterungen in Bezug auf Kameramodelle sei beispielsweise auf [Hartley03] und [Bouguet07] verwiesen. Der folgende Abschnitt wird sich genauer mit den intrinsischen Parametern und der Transformation vom Kamerakoordinatensystem in Bildkoordinaten auseinandersetzen.

Für die in Gleichung 2.16 hergeleiteten intrinsischen Parameter wurde die Verzerrung durch die Kameralinse nicht berücksichtigt, welche jedoch entsteht und korrigiert werden muss. Für die Verzerrung ist die Kenntnis der Verzerrungsparameter von Bedeutung. Für die Projektion eines 3D-Punktes in die Bildebene gilt (vgl. 2.12):

$$x_n = \begin{bmatrix} cx/cz \\ cy/cz \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Betrachtet man nun die Verzerrung  $x_d$  durch die Linse, ergibt sich folgende Gleichung, die sich aus radialer und tangentialer Verzerrung ( $d_r$  und  $d_t$ ) zusammensetzt (vgl. [Bouguet07]):

$$x_d = d_r + d_t \quad (2.17)$$

$$x_d = \begin{bmatrix} (1 + kc_1r^2 + kc_2r^4 + kc_5r^6) \\ (1 + kc_1r^2 + kc_2r^4 + kc_5r^6) \\ 1 \end{bmatrix} x_n + d_t \quad (2.18)$$

Der Radius  $r$  ergibt sich aus den beiden Koordinaten von  $x_n$  zu  $r^2 = x^2 + y^2$ . Die tangentialer Verzerrung  $d_t$  lässt sich wie folgt berechnen:

$$d_t = \begin{bmatrix} 2kc_3xy + kc_4(r^2 + 2x^2) \\ kc_3(r^2 + 2y^2) + 2kc_4xy \\ 1 \end{bmatrix}$$

Die Variablen  $kc_1 \dots kc_5$  beschreiben die Verzerrungsparameter<sup>6</sup>, die nach [Bouguet07] für dieses Modell notwendig sind. Die eigentlichen Bildkoordinaten  ${}^i x$  ergeben sich nun aus den Koordinaten mit der Verzerrung  $x_d$  und der Kalibrierungsmatrix aus Gleichung 2.16 zu:

$${}^i x = K \cdot x_d \quad (2.19)$$

$${}^i x = \begin{bmatrix} f_x & \text{skew} \cdot f_x & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot x_d \quad (2.20)$$

Ausgehend von dieser Gleichung sind für die Bestimmung der intrinsischen Matrix einer Kamera folgende Parameter wichtig:

- Brennweite  $\alpha_x = f_x$  und  $\alpha_y = f_y$  der jeweiligen Achsen in Pixel
- Mittelpunkt  $(x_0, y_0)$  des Bildes
- Skew-Parameter  $s = \text{skew} \cdot f_x$  des Bildes
- Verzerrungsfaktoren  $kc_1 \dots kc_5$

---

<sup>6</sup>engl. *distortion factor*

Für die in dieser Arbeit verwendete CCD-Kamera wurden folgende Parameter ermittelt und verwendet:

$$\begin{array}{lll} f_x = 558,3783783\text{px} & kc_1 = -0,2029 & x_0 = 357,9062\text{px} \\ f_y = 558,2432432\text{px} & kc_2 = 0,1352 & y_0 = 267,2745\text{px} \end{array}$$

Die Berechnung der Verzerrung wurde vereinfacht betrachtet. So wurden die Verzerrungsparameter  $kc_3 \dots kc_5$  als 0 angenommen. Das bedeutet, dass die tangentielle Verzerrung ignoriert und die radiale Verzerrung nur bis zur vierten Ordnung betrachtet wird.

### 2.3.3 Grundlagen der Wahrscheinlichkeitstheorie

In diesem Abschnitt werden Grundlagen der Wahrscheinlichkeitstheorie in Bezug auf die später verwendeten Filter und die Robotertechnik näher erläutert.

Um die Bewegung und Lokalisierung der Roboter verstehen zu können, wird die Unsicherheit ihrer Position mit Hilfe der Wahrscheinlichkeitstheorie beschrieben. Mehrdeutigkeiten können auf mathematischem Wege mit der Wahrscheinlichkeit betrachtet und Stellgrößen der Roboter verarbeitet werden. Mit Hilfe von verschiedensten Wahrscheinlichkeitstheoretischen Algorithmen lässt sich die Unsicherheit des Zustandes eines Roboters reduzieren und zu einem momentanen Zustand zusammenfassen.

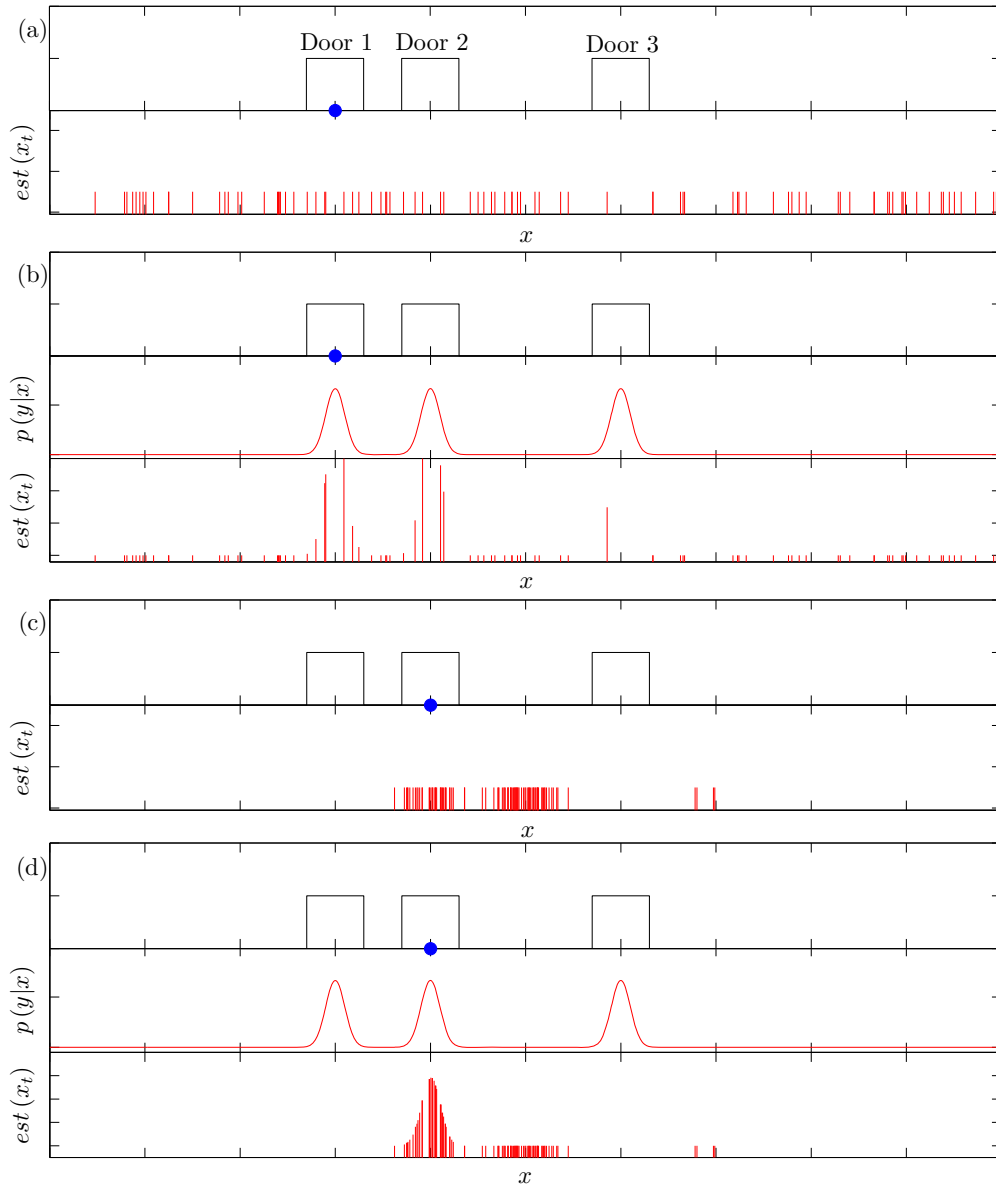
In der Abbildung 2.11 ist die mobile Lokalisierung eines Roboters dargestellt. Das Problem besteht in der Bestimmung der Koordinaten relativ zu einer Ausgangsreferenz. Mit Hilfe einer Karte der Umgebung versucht sich der Roboter anhand verschiedener Sensoren relativ zu dieser Karte zu lokalisieren.

In der Robotertechnik werden Größen wie Messungen, Stellgrößen, der Zustand des Roboters und die Umgebung mit Zufallsvariablen modelliert. Eine Zufallsvariable<sup>7</sup> kann verschiedene Werte entsprechend der Wahrscheinlichkeitstheorie annehmen und beschreibt somit eine reelle Funktion, die auf dem Raum der Elementarereignisse  $\Omega$  (Ereignisraum, Merkmalsraum) definiert ist. Die Werte  $X(\omega)$ , die die Zufallsgröße  $X$  dabei annehmen kann, nennt man *Realisierungsmengen von  $X$* .

$$W = \{x : x = X(\omega), \omega \in \Omega\}$$

---

<sup>7</sup>Eine Zufallsvariable wird häufig auch als Zufallsgröße bezeichnet.



**Abbildung 2.11:** Lokalisierung mittels Partikelfilter: Ein mobiler Roboter der seine Position mittels einer globalen Lokalisierung bestimmt. Der Partikelfilter wird im Kapitel 3 erklärt.

In der Wahrscheinlichkeitstheorie gelten drei Axiome aus denen sich der Wahrscheinlichkeitsbegriff nach Andrei Kolmogorow ableiten lässt. Jedem Ereignis  $\omega$  des Ereignisraumes  $\Omega$  wird eine Wahrscheinlichkeit  $p(\omega)$  zugeordnet, so dass

- $0 \leq p(\omega) \leq 1$
- $\sum_{\Omega} p(\omega) = 1$
- $p(\omega_1 \cup \omega_2 \cup \dots) = p(\omega_1) + p(\omega_2) + \dots$ , wenn  $\omega_i \cap \omega_j = \emptyset$

Man unterscheidet diskrete und stetige Zufallsvariablen. Beispielsweise würde das Werfen einer Münze eine Zufallsvariable, welche die Werte  $X(\omega) = \{Kopf, Zahl\}$  annehmen kann, modellieren. Für diesen diskreten Wahrscheinlichkeitsraum gibt es nur zwei Ereignisse, nämlich *Kopf* oder *Zahl*. Die daraus resultierenden Wahrscheinlichkeiten werden wie folgt angegeben. Die Verteilung einer Zufallsvariable  $X$

$$F(X) = p(X \leq x), \quad -\infty < x < \infty \quad (2.21)$$

wird als *Verteilungsfunktion der Zufallsvariable*  $X^8$  bezeichnet und beschreibt die Wahrscheinlichkeit dafür, dass die Zufallsvariable  $X$  den Realisierungswert kleiner oder gleich  $x$  angenommen hat. So würden sich für den Münzwurf folgende Wahrscheinlichkeiten ergeben:  $p(X = Kopf) = p(X = Zahl) = \frac{1}{2}$ . Im Folgenden wird die Beschreibung der Wahrscheinlichkeit eines Ereignisses  $p(X = x)$  mit  $p(x)$  abgekürzt.

Wahrscheinlichkeiten sind immer nicht-negativ und für Verteilungsfunktionen gilt:

$$F(X) = \sum_k p(X = x_k) = 1 \quad (\text{diskrete Zufallsvariable}) \quad (2.22)$$

$$F(X) = \int p(X = x) dx = 1 \quad (\text{stetige Zufallsvariable}) \quad (2.23)$$

Man spricht von einer stetigen oder kontinuierlichen Zufallsgröße, wenn ihre Verteilungsfunktion  $F(X)$  entsprechend Gleichung 2.23 gegeben ist. Für Zufallsgrößen in der Robotertechnik wird angenommen, dass diese durch Wahrscheinlichkeitsdichtefunktionen<sup>9</sup> beschrieben werden können [Wanka04]. Eine für die weiteren Betrachtungen wichtige Dichtefunktion ist die Normalverteilung. Diese ist für den 1-dimensionalen

<sup>8</sup>engl. *cumulative distribution function (CDF)*

<sup>9</sup>engl. *probability density function (PDF)*

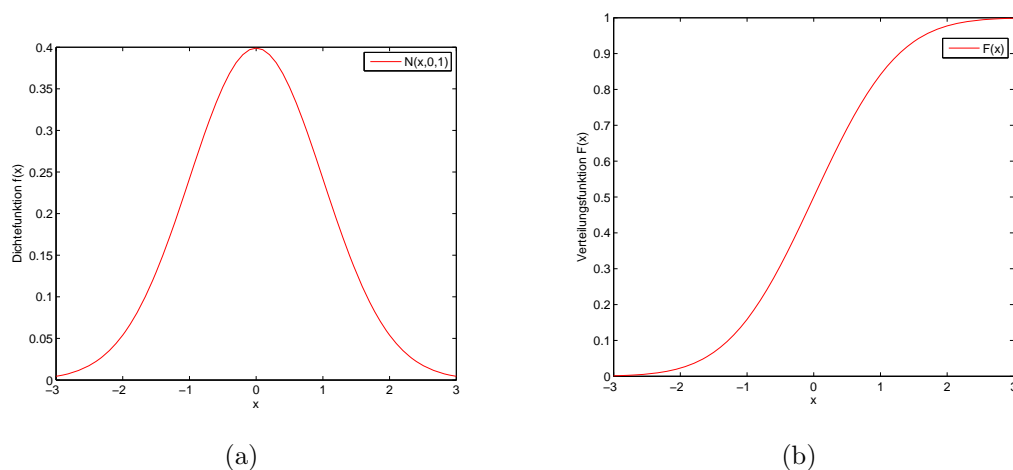
Fall durch den Mittelwert  $\mu$  und die Varianz  $\sigma^2$  folgendermaßen beschrieben:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{1}{2} \cdot \frac{(x-\mu)^2}{\sigma^2}} \quad (2.24)$$

und wird häufig durch  $N(x, \mu, \sigma^2)$  abgekürzt (Abbildung 2.12). Für die 1-dimensionale Normalverteilung ist die Zufallsvariable  $x$  ein Skalar. In der Robotertechnik ist diese häufig ein mehrdimensionaler Vektor. Daraus ergibt sich für eine  $n$ -dimensionale Zufallsvariable  $\underline{x}$  die sogenannte *multivariate Normalverteilung*:

$$p(\underline{x}) = \frac{1}{\sqrt{|2\pi\Sigma|}} \cdot e^{-\frac{1}{2} \cdot (\underline{x}-\underline{\mu})^T \Sigma^{-1} (\underline{x}-\underline{\mu})} \quad (2.25)$$

mit  $\underline{\mu}$  als  $n$ -dimensionalem Erwartungswertvektor und  $\Sigma$  als positiv-semidefinite und symmetrische Kovarianzmatrix.



**Abbildung 2.12:** Dichtefunktion (a) und Verteilungsfunktion (b) einer Normal- oder Gauß-Verteilung

### Bedingte Wahrscheinlichkeit

Als gemeinsame Wahrscheinlichkeit zweier Zufallsvariablen  $X$  und  $Y$  bezeichnet man die Wahrscheinlichkeit, bei der für  $X$  der Wert  $x$  **und** für  $Y$  der Wert  $y$  eintritt:

$$p(x, y) = p(X = x \text{ und } Y = y) \quad (2.26)$$

Desweiteren gilt für zwei Zufallsvariablen  $X$  und  $Y$ , die voneinander unabhängig sind:

$$p(x, y) = p(x) \cdot p(y) \quad (2.27)$$

Es kommt häufig vor, dass Zufallsvariablen voneinander abhängen. Es wird angenommen, dass  $Y$  mit dem Wert  $y$  bekannt ist und die Wahrscheinlichkeit für eine Zufallsvariable  $X$  unter der Bedingung, dass das Ereignis von  $Y$  schon eingetreten ist, gesucht wird. So eine Wahrscheinlichkeit wird als *bedingte Wahrscheinlichkeit* bezeichnet:

$$p(x|y) = p(X = x|Y = y)$$

Ist die Wahrscheinlichkeit  $p(y) > 0$  für das Ereignis  $y$ , so ergibt sich:

$$p(x|y) = \frac{p(x, y)}{p(y)} \quad (2.28)$$

Falls  $X$  und  $Y$  unabhängig sind, folgt daraus:

$$p(x|y) = \frac{p(x)p(y)}{p(y)} = p(x)$$

Anders ausgedrückt ergibt sich für die bedingte Wahrscheinlichkeit zweier unabhängiger Zufallsvariablen  $X$  und  $Y$ , dass  $Y$  keine Information über  $X$  enthält.

### **Totale Wahrscheinlichkeit und der Satz von Bayes**

Als Satz der *totalen Wahrscheinlichkeit* wird die Beziehung zwischen bedingter Wahrscheinlichkeit (vgl. 2.28) und den Axiomen der Wahrscheinlichkeitstheorie bezeichnet. Daraus ergibt sich für die *totale Wahrscheinlichkeit* folgende Formel:

$$p(x) = \sum_k p(x|y_k) p(y_k) \quad (\text{diskret}) \quad (2.29)$$

$$p(x) = \int p(x|y) p(y) dy \quad (\text{stetig}) \quad (2.30)$$

Aus dem Satz der bedingten Wahrscheinlichkeit 2.28 folgt unmittelbar auch der Satz von Bayes. Dieser ergibt sich aus der Wahrscheinlichkeit  $p(x|y)$  und der „Inversen“  $p(y|x)$ :



$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\sum_{k=1}^{\infty} p(y|x_k)p(x_k)} \quad (\text{diskret}) \quad (2.31)$$

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\int_{-\infty}^{\infty} p(y|x_k)p(x_k)dx_k} \quad (\text{stetig}) \quad (2.32)$$

Der Satz von Bayes spielt in der Robotertechnik eine wichtige Rolle. Wenn  $x$  eine Zustandsgröße ist, die aus der Zufallsgröße  $y$  folgt, dann bezeichnet man die Wahrscheinlichkeit  $p(x)$  als *A-priori-Wahrscheinlichkeitsverteilung* und  $y$  als Messung (z.B. Sensordaten). Die Verteilung  $p(x)$  beinhaltet die vorherigen Informationen bezüglich  $X$  und den entsprechenden Messungen  $Y$ . Die Wahrscheinlichkeit  $p(x|y)$  stellt die *A-posteriori-Wahrscheinlichkeitsverteilung* über  $X$  dar. Mit Hilfe des Satzes von Bayes kann man auf einfache Weise  $p(x|y)$  aus der bedingten Wahrscheinlichkeit  $p(y|x)$  und der A-priori-Wahrscheinlichkeit  $p(x)$  berechnen. Anders ausgedrückt, folgt eine Zustandsgröße  $x$  aus den Messdaten  $y$  durch die „inverse“ Wahrscheinlichkeit (*Satz von Bayes*), die die Wahrscheinlichkeit der Messdaten  $y$  unter der Bedingung, dass  $x$  schon eingetreten ist, ausdrückt. Die Wahrscheinlichkeit  $p(y|x)$  beschreibt in der Robotertechnik ein Modell, wie die Zustandsvariable  $X$  die Sensormessungen  $Y$  hervorruft.

Weiterhin lässt sich in Gleichung 2.32 feststellen, dass die Wahrscheinlichkeit  $p(y)$  nicht von  $x$  abhängt. Aus diesem Grund wird häufig  $p(y)^{-1}$  als Normalisierungsfaktor  $\eta$  des Satzes von Bayes bezeichnet:

$$p(x|y) = \eta p(y|x)p(x) \quad (2.33)$$

Für eine weitere Zufallsvariable  $Z$  ergibt sich der Satz von Bayes zu folgender Formel für den Fall, dass  $p(y|z) > 0$  ist:

$$p(x|y, z) = \frac{p(y|x, z)p(x|z)}{p(y|z)} \quad (2.34)$$

Desweiteren gilt für zwei unabhängige Zufallsvariablen  $X, Y$  unter der Bedingung einer Zufallsvariable  $Z$  (vgl. Gleichung 2.27):

$$p(x, y|z) = p(x|z)p(y|z) \quad (2.35)$$

Diese Beziehung wird auch *bedingte Unabhängigkeit* genannt. Diese Aussage ist äquivalent zu folgenden Formeln:

$$p(x|z) = p(x|y, z) \quad (2.36)$$

$$p(y|z) = p(y|x, z) \quad (2.37)$$

Bedingte Unabhängigkeit beschreibt dabei eine Zufallsvariable  $Y$ , die keine Informationen über eine Variable  $X$  enthält, wenn der Wert für  $Z$  bekannt ist. Desweiteren gilt im Allgemeinen nicht, dass aus bedingter Unabhängigkeit auch absolute Unabhängigkeit (vgl. Gleichung 2.27) folgt und umgekehrt:

$$p(x, y|z) = p(x|z)p(y|z) \not\Rightarrow p(x, y) = p(x)p(y)$$

Für viele Algorithmen ist die Kenntnis bestimmter Merkmale einer Wahrscheinlichkeitsdichtefunktion von Bedeutung. Diese Merkmale werden häufig durch *Momente*<sup>10</sup> beschrieben. Eine Normalverteilung ist durch den Erwartungswert und die Kovarianzmatrix festgelegt. Der Erwartungswert einer Zufallsvariable  $X$  beschreibt den Mittelwert einer Wahrscheinlichkeitsverteilung und ergibt sich zu:

$$E(X) = \sum_k x_k p(x_k) \quad (\text{diskret}) \quad (2.38)$$

$$E(X) = \int x p(x) dx \quad (\text{stetig}) \quad (2.39)$$

Die Kovarianz<sup>11</sup> einer 1-dimensionalen Zufallsvariablen  $X$  beschreibt die quadratische Abweichung vom Erwartungswert und entspricht somit der Varianz:

$$\text{Cov}(X) = \text{Var}(X) = E(X - E(X))^2 = E(X^2) - E(X)^2 \quad (2.40)$$

Für einen Zufallsvektor  $\underline{X} = (X_1 \dots X_n)$  beschreibt die Kovarianz den Zusammenhang der Zufallsvariablen  $X_i$  untereinander.

---

<sup>10</sup>Momente sind Kenngrößen der Verteilungsfunktion einer Zufallsvariablen.

<sup>11</sup>Die Kovarianz ist eine Verallgemeinerung der Varianz.

### 3 Filter & Modelle

Um die Umwelt eines Roboters modellieren zu können, muss diese mathematisch erfasst und beschrieben werden können. Die Umgebung eines Roboters sowie der Roboter selbst werden durch ein dynamisches System beschrieben. In diesem System wird mit Hilfe von Zuständen die Umwelt modelliert. Dabei stellt ein Zustand alle Einflüsse des Roboters und der Umgebung dar, die die Zukunft beeinflussen. Somit stellt ein Zustand einen Zufallsvektor dar, der aus mehreren Zufallsvariablen  $X = (X_1 \dots X_n)$  mit  $n \in \mathbb{N}$  auf dem selben Merkmalsraum  $\Omega$  bestehen kann und die Abbildung  $X : \Omega \rightarrow \mathbb{R}^n$  definiert. Der Zustand wird meist als  $x$  bezeichnet und der Zustand zu einem bestimmten Zeitpunkt  $t$  als  $x_t$ . Ein Beispiel für Zustandsvariablen sind:

- die Position eines Roboters, die durch verschiedene Variablen beschrieben werden kann (z.B. x-y-Koordinaten und die Ausrichtung zu einem Referenzsystem)
- die statische Position von Landmarken (vgl. 2.2)

Ein Zustand wird als *vollständig* bezeichnet, wenn dieser als beste Vorhersage für die Zukunft angenommen werden kann. Das bedeutet, alle vergangenen Zustände, Messungen und Stellgrößen beinhalten keine zusätzlichen Informationen, die die Vorhersage verbessern könnten. Temporären Prozesse, die diese Bedingung erfüllen, werden auch als *Markov-Kette* bezeichnet. Die Vollständigkeit eines Zustandes ist jedoch nur von theoretischem Interesse, denn in der Praxis ist es für Roboter unmöglich, einen vollständigen Zustand zu beschreiben, da dieser von zu vielen Ereignissen abhängt. Aus diesem Grund werden meist die Zustände als eine Untermenge aller möglichen Zustände betrachtet. Ein Roboter kann den Zustand seiner Umgebung durch seine Stellgrößen oder seine Sensoren beeinflussen. Die Auswertung der Informationen der Sensoren wird als *Messung* bezeichnet. Diese Messungen liefern einen momentanen Zustand der Umgebung und werden als  $y_t$  bezeichnet. Folgende Gleichung

$$y_{t_1:t_2} = y_{t_1}, y_{t_1+1}, y_{t_1+2} \dots, y_{t_2}$$

beschreibt die Menge aller Messungen, die zwischen dem Zeitpunkt  $t_1$  und  $t_2$  mit  $t_1 \leq t_2$  gemessen wurden. Als Stellgrößen werden Informationen bezeichnet, die den Zustand der Umgebung eines Roboters ändern. Beispielsweise wird meist die Geschwindigkeit des Roboters als Stellgröße dargestellt. Andererseits gibt es auch Anwendungen die ohne Stellgrößen auskommen und den Zustand nur mit Hilfe von Messungen bestimmen. Stellgrößen werden mit  $u_t$  bezeichnet und korrespondieren zu der Änderung in dem Zeitintervall  $(t - 1; t]$ . Eine Folge von Stellgrößen  $u_{t_1:t_2}$  mit  $t_1 \leq t_2$  wird wie folgt bezeichnet:

$$u_{t_1:t_2} = u_{t_1}, u_{t_1+1}, u_{t_1+2} \dots, u_{t_2}$$

Im Allgemeinen wird der Zustand  $x_t$  zu einem Zeitpunkt  $t$  stochastisch von den vergangenen Zuständen, Messungen und Stellgrößen abgeleitet. Aus diesem Grund ergibt sich für den Zustand  $x_t$  die Wahrscheinlichkeitsverteilung  $p(x_t | x_{0:t-1}, y_{1:t-1}, u_{1:t})$ . Für den Fall, dass der Zustand  $x_t$  vollständig ist, also einem *Markov-Prozess* beschreibt, ist der vorherige Zustand  $x_{t-1}$  ausreichend, um alle vergangenen Messungen und Stellgrößen zu beschreiben. Durch bedingte Unabhängigkeit (Gleichung 2.35) gilt:

$$p(X = x_t | X = x_{0:t-1}, Y = y_{1:t-1}, U = u_{1:t}) = p(x_t | x_{t-1}, u_t) \quad (3.1)$$

Die Wahrscheinlichkeit  $p(x_t | x_{t-1}, u_t)$  wird auch als *Zustandsübergangswahrscheinlichkeit* bezeichnet. Diese beschreibt, wie der Zustand  $x_t$  über die Zeit aus den Stellgrößen und dem vorherigen Zustand folgt. Desweiteren lässt sich das gleiche Prinzip auf die Vorhersage von Messungen unter der Bedingung, dass der Zustand  $x_t$  eingetreten ist, anwenden:

$$p(Y = y_t | X = x_{0:t}, Y = y_{1:t-1}, U = u_{1:t}) = p(y_t | x_t) \quad (3.2)$$

Diese Wahrscheinlichkeit wird auch als *Messwahrscheinlichkeit* oder *Messübergangswahrscheinlichkeit* bezeichnet. Dabei beschreibt diese die Wahrscheinlichkeit einer Messung  $y_t$  unter der Bedingung, dass der Zustand  $x_t$  schon eingetreten ist. Die beiden Wahrscheinlichkeiten (*Zustandsübergangs-* und *Messwahrscheinlichkeit*) beschreiben ein dynamisches stochastisches System, welches auch als „*hidden Markov model*“ oder „*dynamic Bayes network*“ bezeichnet wird. In der Robotertechnik ist es wichtig, den Systemzustand eines Roboters zu bestimmen. Da dieser Zustand nicht direkt messbar ist, sondern von verfügbaren Daten abgeleitet werden muss, wird

dieser Systemzustand durch eine Wahrscheinlichkeitsverteilung (*Schätzung*) repräsentiert. Diese Verteilung spiegelt das Wissen über den Zustand der Umgebung eines Roboters wieder. Aus diesem Grund wird der „wahre“ Zustand eines Roboters von der Schätzung<sup>1</sup> dieses Zustandes unterschieden. Desweiteren werden solche Verteilungen durch A-posteriori-Wahrscheinlichkeiten einer Zustandsvariable bedingt von den verfügbaren Messungen und Stellgrößen dargestellt. Für eine Zustandsvariable  $x_t$  wird eine Zustandsverteilung mit  $est(x_t)$  bezeichnet und stellt eine Vereinfachung der A-posteriori-Wahrscheinlichkeit dar:

$$est(x_t) = p(x_t | y_{1:t}, u_{1:t}) \quad (3.3)$$

Diese A-posteriori-Wahrscheinlichkeit ist eine Verteilung für den Zustand  $x_t$  zum Zeitpunkt  $t$  bedingt durch alle vergangenen Messungen  $y_{1:t}$  und Stellgrößen  $u_{1:t}$ . Die Wahrscheinlichkeit vor Verarbeitung der letzten Messung  $y_t$  wird als „Vorhersage“<sup>2</sup> bezeichnet:

$$\widetilde{est}(x_t) = p(x_t | y_{1:t-1}, u_{1:t}) \quad (3.4)$$

Die Wahrscheinlichkeit  $\widetilde{est}(x_t)$  bestimmt den Zustand zum Zeitpunkt  $t$  nur aufgrund vorheriger A-posteriori-Wahrscheinlichkeiten, bevor die eigentliche Messung  $y_t$  verarbeitet wurde. Desweiteren bezeichnet man die Berechnung der Wahrscheinlichkeit  $est(x_t)$  aus  $\widetilde{est}(x_t)$  als Korrektur-<sup>3</sup> oder Updateschritt<sup>4</sup>. Weiterführende Informationen sind in [Thrun05] zu finden.

## 3.1 Bayes-Filter

Einer der wichtigsten Filter-Algorithmen ist der Bayes-Filter. Dieser Filter stellt die Grundlage für viele weitere Filter, wie beispielsweise den *Kalman-Filter* und den *Partikel-Filter*, dar. Der Bayes-Algorithmus leitet sich direkt aus dem Bayes-Theorem (Gleichung 2.32) ab und berechnet den Systemzustand  $est(x_t)$  aus Messdaten und Stellgrößen.

In Tabelle 3.1 ist dieser Algorithmus für einen kontinuierlichen Zustandsraum mit

---

<sup>1</sup>engl. *estimation*

<sup>2</sup>engl. *prediction*

<sup>3</sup>engl. *correction*

<sup>4</sup>engl. *measurement update*

seinen zwei wesentlichen Schritten als Pseudo-Code dargestellt. Der Bayes-Filter berechnet rekursiv den Systemzustand  $est(x_t)$  zum Zeitpunkt  $t$  aus dem vorherigen Zustand  $est(x_{t-1})$ . Im ersten Schritt (Zeile 3) wird der vorhergesagte Systemzustand  $\widetilde{est}(x_t)$  basierend auf dem vorherigen Zustand  $est(x_{t-1})$  und den Stellgrößen  $u_t$  ermittelt. Genauer gesagt, entspricht  $\widetilde{est}(x_t)$  dem Integral des Produktes zweier Wahrscheinlichkeitsverteilungen (vgl. Gleichung 2.30). Wie bereits erwähnt, wird dieser Schritt als Vorhersage (*prediction*) bezeichnet.

```

1 Algorithmus Bayes_Filter( $est(x_{t-1}), u_t, y_t$ ):
2   for all  $x_t$  do
3      $\widetilde{est}(x_t) = \int p(x_t|u_t, x_{t-1}) est(x_{t-1}) dx_{t-1}$ 
4      $est(x_t) = \eta p(y_t|x_t) \widetilde{est}(x_t)$ 
5   endfor
6   return  $est(x_t)$ 

```

**Tabelle 3.1:** Allgemeiner Algorithmus des Bayes-Filters für einen kontinuierlichen Zustandsraum

Für den zweiten Schritt in Zeile 4 wird der vorhergesagte Systemzustand  $\widetilde{est}(x_t)$  mit der Wahrscheinlichkeit, dass die Messung  $y_t$  beobachtet worden ist, multipliziert. Dieser Schritt wird auch als Korrekturschritt (*measurement update*) bezeichnet. Das Ergebnis wird mit Hilfe der Konstante  $\eta$  normalisiert und zurückgegeben (Zeile 6). Um die A-posteriori-Wahrscheinlichkeit berechnen zu können, benötigt der Algorithmus einen Initialzustand  $est(x_0)$  zum Zeitpunkt  $t = 0$  als Randbedingung.

Tabelle 3.2 stellt den Bayes-Filter für einen finiten Zustandsraum dar. Dabei unterscheidet sich dieser Algorithmus nur hinsichtlich der Ersetzung der Integration durch eine finite Summe. Die Variablen  $x_{k,t}$  und  $x_{i,t}$  beschreiben hierbei individuelle Zustände, von denen es nur eine bestimmte Anzahl gibt. Die Schätzung  $est(x_t)$  zum Zeitpunkt  $t$  wird durch die Wahrscheinlichkeit  $p(x_{k,t})$  jedes Zustandes  $x_{k,t}$  beschrieben. Die Prädiktion (Tabelle 3.2 Zeile 3) und die Korrektur (Tabelle 3.2 Zeile 4) werden entsprechend des kontinuierlichen Bayes-Filters durchgeführt. Weiterführende Informationen bezüglich des Bayes-Filters sind in [Thrun05] zu finden.

```

1 Algorithmus Bayes_Filter_Discrete({p(xk,t-1)}, ut, yt):
2   for all k do
3      $\tilde{p}(x_{k,t}) = \sum_i p(X_t = x_k | u_t, X_{t-1} = x_i) p(x_{i,t-1})$ 
4      $p(x_{k,t}) = \eta p(Y_t = y_t | X_t = x_k) \tilde{p}(x_{k,t})$ 
5   endfor
6   return {p(xk,t)}

```

**Tabelle 3.2:** Allgemeiner Algorithmus des Bayes-Filters für einen finiten Zustandsraum

### 3.1.1 Mathematische Betrachtungen des Bayes-Filters

Unter der Annahme, dass die Schätzung  $est(x_0)$  korrekt initialisiert wurde, der Zustand  $x_t$  vollständig ist und die Stellgrößen willkürlich ausgewählt werden, wird gezeigt, dass sich die A-posteriori-Wahrscheinlichkeitsverteilung  $p(x_t | y_{1:t}, u_{1:t})$  aus der Verteilung  $est(x_{t-1})$  zum Zeitpunkt  $t - 1$  berechnen lässt [Thrun05]. Die gesuchte Schätzung  $p(x_t | y_{1:t}, u_{1:t})$  ergibt sich aus dem Satz von Bayes (Gleichung 2.34) wie folgt:

$$est(x_t) = p(x_t | y_{1:t}, u_{1:t}) = \frac{p(y_t | x_t, y_{1:t-1}, u_{1:t}) p(x_t | y_{1:t-1}, u_{1:t})}{p(y_t | y_{1:t-1}, u_{1:t})} \quad (3.5)$$

$$= \eta \underbrace{p(y_t | x_t, y_{1:t-1}, u_{1:t})}_{\text{Messgleichung}} \underbrace{p(x_t | y_{1:t-1}, u_{1:t})}_{\tilde{est}(x_t)} \quad (3.6)$$

Unter der Annahme eines *Markov*-Prozesses und der Vollständigkeit des Zustands  $x_t$  (vgl. Abschnitt 3) ergibt sich aus der bedingten Unabhängigkeit (Gleichung 2.35):

$$p(y_t | x_t, y_{1:t-1}, u_{1:t}) = p(y_t | x_t) \quad (3.7)$$

Dabei werden die vergangenen Messungen  $y_{1:t-1}$  und die Stellgrößen  $u_{1:t}$  nicht weiter betrachtet. Mit anderen Worten bedeutet dies, dass für die Prädiktion der Messung  $y_t$  durch den Zustand  $x_t$  vergangene Messungen und Stellgrößen keine zusätzlichen Informationen liefern. Dadurch ergibt sich vereinfacht für Gleichung 3.6:

$$p(x_t | y_{1:t}, u_{1:t}) = \eta p(y_t | x_t) p(x_t | y_{1:t-1}, u_{1:t}) \quad (3.8)$$

Ersetzt man die Messgleichung  $p(y_t|x_t, y_{1:t-1}, u_{1:t}) = p(y_t|x_t)$  und die vorhergesagte Schätzung  $p(x_t|y_{1:t-1}, u_{1:t}) = \widetilde{est}(x_t)$ , so folgt:

$$est(x_t) = \eta p(y_t|x_t) \widetilde{est}(x_t) \quad (3.9)$$

Diese Gleichung ist in Tabelle 3.1 in Zeile 4 dargestellt. Als nächstes wird die vorhergesagte Schätzung  $\widetilde{est}(x_t)$  wie folgt entsprechend Gleichung 2.30 erweitert:

$$\begin{aligned} \widetilde{est}(x_t) &= p(x_t|y_{1:t-1}, u_{1:t}) \\ &= \int \underbrace{p(x_t|x_{t-1}, y_{1:t-1}, u_{1:t})}_{\text{Zustandsübergangsgleichung}} \underbrace{p(x_{t-1}|y_{1:t-1}, u_{1:t})}_{est(x_{t-1})} dx_{t-1} \end{aligned}$$

Erneut wird der Zustand  $x_t$  als vollständig betrachtet. Das bedeutet, dass durch die Kenntnis des vorherigen Zustandes  $x_{t-1}$ , die vorherigen Messungen  $y_{1:t-1}$  und die Stellgrößen  $u_{1:t-1}$  keinen Einfluss auf den Zustand  $x_t$  haben und folglich in der Gleichung entfallen können. Dies führt zu folgender Gleichung:

$$p(x_t|x_{t-1}, y_{1:t-1}, u_{1:t}) = p(x_t|x_{t-1}, u_t) \quad (3.10)$$

Die Stellgröße  $u_t$  wird in der Betrachtung beibehalten, da nur diese letzte Stellgröße Einfluss auf den Zustand  $x_t$  hat. Durch Ersetzen der vorherigen Schätzung  $p(x_{t-1}|y_{1:t-1}, u_{1:t-1}) = est(x_{t-1})$  führt dies zu der in Tabelle 3.1 Zeile 3 dargestellten Zustandsübergangsgleichung:

$$\widetilde{est}(x_t) = \int p(x_t|x_{t-1}, u_t) est(x_{t-1}) dx_{t-1} \quad (3.11)$$

## 3.2 Kalman-Filter

Der Kalman-Filter ist ein stochastischer Systemzustandsschätzer für dynamische Systeme. Im Jahre 1960 wurde dieser von Rudolf Kalman erstmals für zeitdiskrete, lineare Gauß'sche Systeme entwickelt. Mit Hilfe dieses Filters ist es möglich, den Zustand eines Systems aufgrund von redundanten verrauschten Messungen zu schätzen.

Der Kalman-Filter charakterisiert seinen Systemzustand für zeitdiskrete, lineare Systeme durch Gauß'sche Normalverteilungen. Der Filter wird durch die sogenannte



*Zustandsübergangsgleichung* und die *Messgleichung* bestimmt.

Der Kalman-Filter repräsentiert seinen momentanen Systemzustand  $est(x_t)$  über die Normalverteilung mit dem Erwartungswert  $\hat{x}$  und der Kovarianz  $P$ . Dabei werden die Zustandsvektoren  $x_t$  und  $x_{t-1}$  zum Zeitpunkt  $t$  mit  $x \in \mathbb{R}^n$  und der Stellgrößenvektor  $u_t$  mit  $u \in \mathbb{R}^m$  beschrieben. Die Zustandsübergangsgleichung ist für den Kalman-Filter eine *lineare Funktion* und wird wie folgt angegeben:

$$x_t = A_t x_{t-1} + B_t u_t + w_t \quad (3.12)$$

Die Zustandsübergangsmatrix  $A_t$  ist eine quadratische Matrix der Größe  $n \times n$ , wobei  $n$  die Dimension des Zustandsvektors  $x_t$  darstellt und überführt den Zustand  $x_{t-1}$  in einen neuen Zustand  $x_t$ . Die Eingangsmatrix  $B_t$  ist eine Matrix der Größe  $n \times m$ , wobei  $m$  die Dimension des Stellgrößenvektors  $u_t$  ist und erlaubt somit den Einfluss von Stellgrößen in das System. Die Zufallsvariable  $w_t$  ist ein normalverteilter Vektor, der das Prozessrauschen modelliert. Für diesen Vektor wird als Mittelwert *null* und als Kovarianz  $Q_t$  angenommen.

Die Zustandsübergangswahrscheinlichkeit  $p(x_t | u_t, x_{t-1})$  berechnet sich durch Gleichung 3.1 eingesetzt in Gleichung 2.25 und entspricht dem Prädiktionsschritt des Bayes-Filters (Tabelle 3.1 Zeile 3) [Thrun05]:

$$p(x_t | x_{t-1}, u_t) = \frac{1}{\sqrt{|2\pi Q_t|}} \cdot e^{-\frac{1}{2} \cdot (x_t - A_t x_{t-1} - B_t u_t)^T Q_t^{-1} (x_t - A_t x_{t-1} - B_t u_t)} \quad (3.13)$$

Die Messübergangsgleichung wird durch folgende Formel dargestellt und beschreibt ebenfalls eine lineare Funktion:

$$y_t = C_t x_t + \nu_t \quad (3.14)$$

Dabei führt die Messübergangsmatrix  $C_t$  den Zustand  $x_t$  in den Messvektor  $y_t$  über.  $\nu_t$  stellt hierbei das normalverteilte Messrauschen mit dem Mittelwert *null* und der Kovarianz  $R_t$  dar. Daraus ergibt sich für die Messübergangswahrscheinlichkeit folgende Formel, die dem Korrekturschritt des Bayes-Filters (Tabelle 3.1 Zeile 4) entspricht:

$$p(y_t | x_t) = \frac{1}{\sqrt{|2\pi R_t|}} \cdot e^{-\frac{1}{2} \cdot (y_t - C_t x_t)^T R_t^{-1} (y_t - C_t x_t)} \quad (3.15)$$

Desweiteren ist ein Initialzustand  $est(x_0)$  für den Kalman-Filter notwendig, derals normalverteilt vorausgesetzt wird.

```

1 Algorithmus Kalman_Filter( $\hat{x}_{t-1}, P_{t-1}, u_t, y_t$ ):
2    $\tilde{x}_t = A_t \hat{x}_{t-1} + B_t u_t$ 
3    $\tilde{P}_t = A_t P_{t-1} A_t^T + Q_t$ 
4
5    $K_t = \tilde{P}_t C_t^T (C_t \tilde{P}_t C_t^T + R_t)^{-1}$ 
6
7    $\hat{x}_t = \hat{x}_t + K_t (y_t - C_t \tilde{x}_t)$ 
8    $P_t = (I - K_t C_t) \tilde{P}_t$ 
9   return  $\hat{x}_t, P_t$ 

```

**Tabelle 3.3:** Kalman-Filter-Algorithmus für lineare Gauß'sche Systeme mit  $\bar{x}$  als Mittelwertvektor und  $P$  als Kovarianzmatrix [Thrun05]

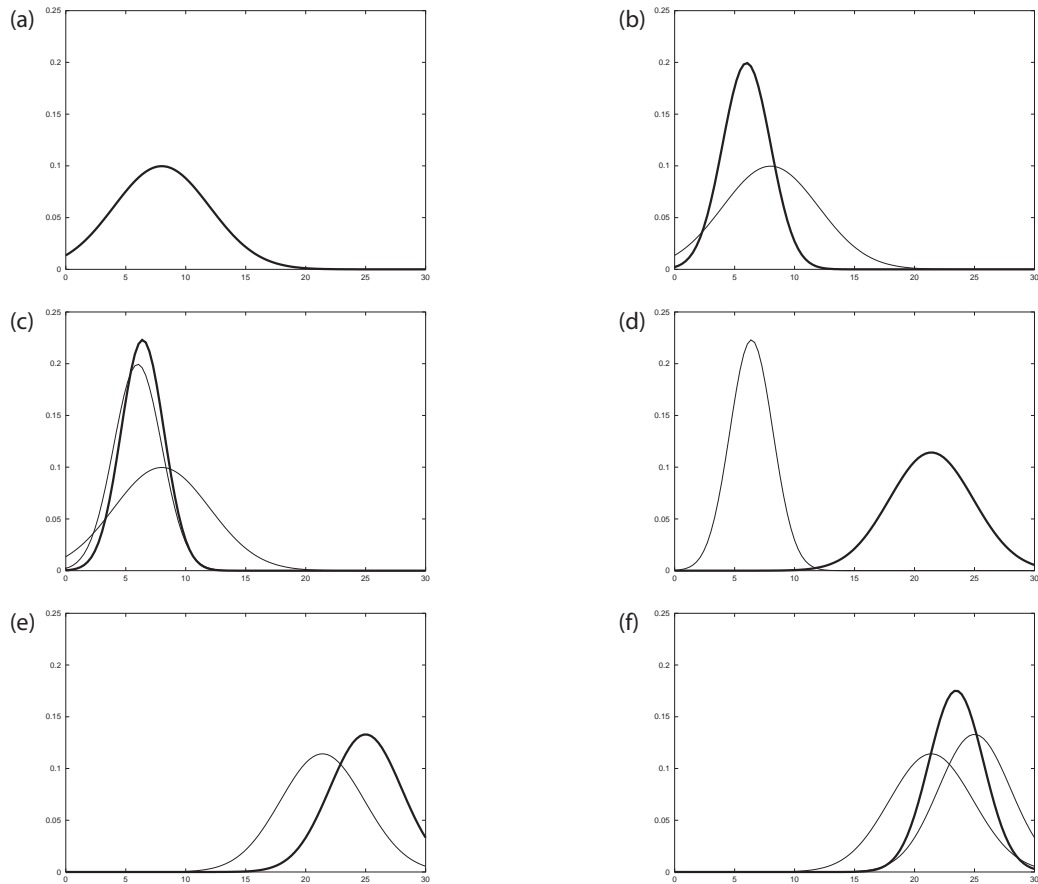
Der Kalman-Filter-Algorithmus beschreibt den Systemzustand  $est(x_t)$  zum Zeitpunkt  $t$  durch den Erwartungswert  $\hat{x}_t$  und die Kovarianz  $P_t$  (Tabelle 3.3). Durch den vorherigen Zustand, charakterisiert durch  $\hat{x}_{t-1}$  und  $P_{t-1}$  und den Stellgrößenvektor  $u_t$  wird der prädiertem Zustand  $\tilde{x}_t$  und  $\tilde{P}_t$  zum Zeitpunkt  $t$  berechnet (Zeile 2-3). Dieser Schritt wird, wie bereits erwähnt, als Prädiktionsschritt bezeichnet und berechnet die vorhergesagte Zustandsverteilung  $\tilde{est}(x_t)$  durch folgende Formel:

$$\tilde{est}(x_t) = \int \underbrace{p(x_t | x_{t-1}, u_t)}_{\sim N(x_t, A_t x_{t-1} + B_t u_t, Q_t)} \underbrace{est(x_{t-1})}_{\sim N(x_{t-1}, \hat{x}_{t-1}, P_{t-1})} dx_{t-1}$$

Der vorhergesagte Zustand  $\tilde{est}(x_t)$  wird in den neuen Systemzustand  $est(x_t)$  transformiert (Zeile 5-8). Dabei wird mit Hilfe des Faktors  $K_t$ , dem sogenannten Kalman-Faktor, die Messung  $y_t$  mit der erwarteten Messung  $C_t \tilde{x}_t$  gewichtet und zu einem neuen Zustand zusammengefasst. Die neue Zustandsverteilung  $est(x_t)$  wird durch den Korrekturschritt (Zeile 7-8) bestimmt und durch folgende Formel ausgedrückt:

$$est(x_t) = \eta \underbrace{p(y_t | x_t)}_{\sim N(y_t, C_t x_t, R_t)} \underbrace{\tilde{est}(x_t)}_{\sim N(x_t, \tilde{x}_t, \tilde{P}_t)}$$

Für weiterführende Informationen und mathematische Erläuterungen bezüglich des Kalman-Filters für lineare Systeme sei auf [Richter05] und [Thrun05] verwiesen.



**Abbildung 3.1:** Veranschaulichung des Kalman-Filters: (a) Initialzustand, (b) eine Messung (fett dargestellt) mit der dazugehörigen Unsicherheit, (c) der Systemzustand, der mit Hilfe des Kalman-Filters durch die Messung bestimmt wurde, (d) der Systemzustand nach einer Bewegung nach rechts (beinhaltet Unsicherheit), (e) eine neue Messung mit dazugehöriger Unsicherheit und (f) der sich ergebende Systemzustand [Thrun05]

### 3.2.1 Kalman-Filter für nichtlineare Systeme

Die Annahme, dass die Zustands- und die Messübergangsgleichung lineare Funktionen repräsentieren, genügt in der Praxis vielen Modellen nicht. Für die meisten Modelle müssen nichtlineare Beziehungen sowohl zwischen dem aktuellen und dem vorhergesagten Zustand als auch zwischen der Messung und dem Zustand vorausgesetzt werden, um physikalische Zusammenhänge möglichst genau abbilden zu können.

Der Erweiterte-Kalman-Filter<sup>5</sup> arbeitet mit nicht-linearen Funktionen für die

<sup>5</sup>engl. *extended Kalman filter (EKF)*

Zustands- und Messübergangsgleichung. Somit folgt aus Gleichung 3.12

$$x_t = g_A(x_{t-1}, u_t) + w_t \quad (3.16)$$

die Zustandsübergangsgleichung und aus Gleichung 3.14

$$y_t = g_C(x_t) + v_t \quad (3.17)$$

die Messübergangsgleichung. Die Funktion  $g_A$  ersetzt die Matrizen  $A_t$  und  $B_t$  in Gleichung 3.12 und die Funktion  $g_C$  die Matrix  $C_t$  in Gleichung 3.14. Aufgrund der nicht-linearen Funktionen  $g_A$  und  $g_C$  stellt die tatsächliche Zustandsverteilung keine geschlossene Lösung mehr dar. Der Grundgedanke des EKF ist die Näherung der nicht-linearen Funktionen durch Linearisierung. Dabei linearisiert dieser die nicht-linearen Anteile mit Hilfe der Taylor-Approximation (erste Ordnung).

Für die Approximation ergibt sich für die nicht-lineare Funktion  $g_A$  folgende Formel:

$$g_A(x_{t-1}, u_t) \approx g_A(\hat{x}_{t-1}, u_t) + \underbrace{g'_A(\hat{x}_{t-1}, u_t)}_{=: J_{A,t}} (x_{t-1} - \hat{x}_{t-1}) \quad (3.18)$$

Der Anstieg der Approximation nach Taylor (erste Ordnung) berechnet sich aus der partiellen Ableitung:

$$J_{A,t} = \frac{\partial g_A(x_{t-1}, u_t)}{\partial x_{t-1}} \quad (3.19)$$

Dabei stellt  $J_{A,t}$  eine  $n \times n$ -Matrix, mit  $n$  als Dimension des Zustandes  $x$ , dar und wird als *Jakobimatrix* bezeichnet. Aus dieser Linearisierung ergibt sich nun die Zustandsübergangswahrscheinlichkeit wie folgt:

$$p(x_t | x_{t-1}, u_t) = \frac{1}{\sqrt{|2\pi Q_t|}} \cdot e^{-\frac{1}{2} \cdot (x_t - g_A(x_{t-1}, u_t))^T Q_t^{-1} (x_t - g_A(x_{t-1}, u_t))} \quad (3.20)$$

Dabei ist  $g_A(x_{t-1}, u_t)$  die Linearisierung aus Gleichung 3.18. Für die Verteilung des vorhergesagten Zustandes  $\widetilde{est}(x_t)$  folgt aus 3.20:

$$\widetilde{est}(x_t) = \int \underbrace{p(x_t | x_{t-1}, u_t)}_{\sim N(x_t, g_A(\hat{x}_{t-1}, u_t) + J_{A,t}(x_{t-1} - \hat{x}_{t-1}), Q_t)} \underbrace{est(x_{t-1})}_{\sim N(x_{t-1}, \hat{x}_{t-1}, P_{t-1})} dx_{t-1}$$

Für die Messübergangsgleichung ergibt sich nach der Linearisierung der Funktion  $G_C$  nach dem Prinzip von 3.18 mit  $J_{C,t} = g'_C = \frac{\partial g_C(x_t)}{\partial x_t}$ :

$$p(y_t|x_t) = \frac{1}{\sqrt{|2\pi R_t|}} \cdot e^{-\frac{1}{2} \cdot (y_t - g_C(\hat{x}_t) - J_{C,t}(x_t - \hat{x}_t))^T R_t^{-1} (y_t - g_C(\hat{x}_t) - J_{C,t}(x_t - \hat{x}_t))} \quad (3.21)$$

Somit wird die Zustandsverteilung  $est(x_t)$  über den Korrekturschritt (Tabelle 3.4 Zeile 7-8) folgendermaßen bestimmt:

$$est(x_t) = \eta \underbrace{p(y_t|x_t)}_{\sim N(y_t, g_C(\tilde{x}_t) + J_{C,t}(x_t - \tilde{x}_t), R_t)} \underbrace{\tilde{est}(x_t)}_{\sim N(x_t, \tilde{P}_t)}$$

1	Algorithmus Extended_Kalman_Filter( $\hat{x}_{t-1}, P_{t-1}, u_t, y_t$ ):
2	$\tilde{x}_t = g_A(\hat{x}_{t-1}, u_t)$
3	$\tilde{P}_t = J_{A,t} P_{t-1} J_{A,t}^T + Q_t$
4	
5	$K_t = \tilde{P}_t J_{C,t}^T (J_{C,t} \tilde{P}_t J_{C,t}^T + R_t)^{-1}$
6	
7	$\hat{x}_t = \tilde{x}_t + K_t (y_t - g_C(\tilde{x}_t))$
8	$P_t = (I - K_t J_{C,t}) \tilde{P}_t$
9	return $\hat{x}_t, P_t$

**Tabelle 3.4:** Erweiterter-Kalman-Filter-Algorithmus für nicht-lineare Gauß'sche Systeme mit  $\hat{x}$  als Mittelwertvektor und  $P$  als Kovarianzmatrix

Der EKF-Algorithmus unterscheidet sich nicht wesentlich von dem des Kalman-Filters. Die wichtigsten Unterschiede sind in Tabelle 3.5 zusammengefasst. Durch die nicht-linearen Abhängigkeiten benutzt der EKF die Jakobimatrizen  $J_{A,t}$  und  $J_{C,t}$  anstatt der korrespondierenden linearen Matrizen  $A_t$ ,  $B_t$  und  $C_t$  des Kalman-Filters. Für weiterführende Informationen bezüglich des EKF und dazugehöriger mathematischer Zusammenhängen sei auf [Richter05] und [Thrun05] verwiesen.

### 3.2.2 Systemmodell

Als Systemmodell versteht man die Beschreibung der Zustandsübergangswahrscheinlichkeit  $p(x_t|x_{t-1}, u_t)$  durch ein geeignetes Modell. Das Modell kann entweder lineare

	KALMAN-FILTER	EKF
Zustandsübergangsgleichung	$A_t \hat{x}_{t-1} + B_t u_t$	$g_A(\hat{x}_{t-1}, u_t)$
Messgleichung	$C_t \tilde{x}_t$	$g_C(\tilde{x}_t)$

**Tabelle 3.5:** Unterschiede der Algorithmen des Kalman-Filters und des Erweiterten-Kalman-Filters.

(vgl. Gleichung 3.12) oder nicht-lineare (vgl. Gleichung 3.16) Zusammenhänge aufweisen. Dabei modelliert das Systemmodell entsprechend der Aufgabe physikalische Zusammenhänge für die Bestimmung der Position eines Roboters und fasst diese in einem Bewegungsmodell zusammen. Da ein solches Modell auf bestimmten Annahmen beruht, kann es die Realität nicht exakt abbilden. Der dadurch entstehende Fehler wird von dem Kalman-Filter berücksichtigt und durch das Prozessrauschen mittels der Kovarianzmatrix  $Q$  modelliert. Es existieren verschiedene Bewegungsmodelle, um die Bewegung eines Roboters zu modellieren. Das in der Aufgabe verwendete Modell wird als „constant turn rate and velocity“-Modell (CTRV-Modell) bezeichnet. Auf Grund der Annahme, dass ein Roboter eine gleichförmig beschleunigte Bewegung in alle Richtungen ausführt, entspricht das Modell einer Bewegung mit der Geschwindigkeit  $v$  und der Winkelbeschleunigung  $\omega$  auf einer Kreisbahn mit konstantem Radius für einen Zeitschritt  $T$  [Richter05].

Der Zustandsvektor  $x_t$  beschreibt Zustandsgrößen, die das System abbilden und ist wie folgt für das CTRV-Modell angegeben:

$$x = [{}^w x, {}^w y, {}^w \gamma, v, \omega]^T \quad (3.22)$$

Die Zustandsgrößen  ${}^w x$  und  ${}^w y$  stellen dabei die Position in einem Weltkoordinatensystem  $w$  dar. Der Winkel  ${}^w \gamma$  entspricht dem Drehwinkel des Roboters bezogen auf die  $x$ -Achse des Weltkoordinatensystem  $w$ . Das CTRV-Modell beschreibt aufgrund der kreisförmigen Bewegung ein nicht-lineares System und wird mit Hilfe von Gleichung 3.16 abgebildet.

Der in der Aufgabe verwendete Erweiterte-Kalman-Filter besitzt keine Stellgrößen  $u_t = 0$ . Somit ergibt sich für 3.16 mit dem Zustandsvektor 3.22 folgende Abhängigkeit:

$$x_t = g_A(x_{t-1}) + w_t$$

Unter Berücksichtigung der Linearisierung (vgl. Gleichung 3.18) ergibt sich für Tabelle 3.4 Zeile 2:

$$\begin{aligned} \tilde{x}_t &= g_A(\hat{x}_{t-1}) \\ &= \begin{bmatrix} \frac{v_{t-1}}{\omega_{t-1}} [\sin(\omega_{t-1}T + {}^w\gamma_{t-1}) - \sin({}^w\gamma_{t-1})] + {}^w x_{t-1} \\ \frac{v_{t-1}}{\omega_{t-1}} [-\cos(\omega_{t-1}T + {}^w\gamma_{t-1}) + \cos({}^w\gamma_{t-1})] + {}^w y_{t-1} \\ \omega_{t-1}T + {}^w\gamma_{t-1} \\ v_{t-1} \\ \omega_{t-1} \end{bmatrix} \end{aligned} \quad (3.23)$$

Dabei beschreibt das Modell eine Diskretisierung  $T = t_{k+1} - t_k$  des kontinuierlichen Systems für gleichförmige Kreisbewegungen mit konstanter Geschwindigkeit (Betrag) und Winkelgeschwindigkeit. Dieses lässt sich folgendermaßen darstellen:

$$x_{t_{k+1}} = \int_{t_k}^{t_{k+1}} \dot{x}_t dt = \int_{t_k}^{t_{k+1}} \begin{bmatrix} v \cos(\omega t) \\ v \sin(\omega t) \\ \omega \\ 0 \\ 0 \end{bmatrix} dt$$

Hierbei beschreiben die Größen  $x_{t_k}$  und  $x_{t_{k+1}}$  die kontinuierliche Zustandsvariable  $x(t)$  eines Systems für gleichförmige Kreisbewegung zu einem bestimmten Zeitpunkt  $t_k$  und  $t_{k+1}$ . Diese Gleichung bezieht sich auf das Koordinatensystem, indem sich der Roboter befindet (Fahrzeugkoordinatensystem). Durch Transformation in das Weltkoordinatensystem entsteht die oben angegebene Beziehung (Gleichung 3.23). Eine genaue Herleitung dieses Modells ist in [Richter05] zu finden.

Für die Geradeausfahrt eines Roboters mit  $\omega = 0$  ist als Spezialfall von 3.23 zu betrachten:

$$\tilde{x}_{t,\omega \rightarrow 0} = \lim_{\omega \rightarrow 0} g_A(\hat{x}_{t-1}) = \begin{bmatrix} \cos({}^w\gamma) vT + {}^w x \\ \sin({}^w\gamma) vT + {}^w y \\ {}^w\gamma \\ v \\ 0 \end{bmatrix}$$

Mit Hilfe der Gleichung 3.23 wird der Zustand  $est(x_{t-1})$  in den vorhergesagten Zustand  $\widetilde{est}(x_t)$  überführt. Für die Linearisierung an der Stelle  $x_{t-1}$  ist außerdem die Bestimmung der Jakobimatrix  $J_{A,t}$  notwendig. Die Jakobimatrix wird nach Gleichung 3.19 bestimmt. Auf die Darstellung dieser Matrix wird jedoch aus Platzgründen verzichtet. Somit lässt sich auch die Kovarianz  $P$  in Tabelle 3.4 Zeile 3 berechnen. Für die Bestimmung der diskreten Systemfehlerkovarianzmatrix  $Q_t$  kann die Formel nach [Scheunert06] und [Richter05] verwendet werden:

$$Q_t = \int_0^T e^{-J_{A_c}(T-\tau)} Q^c [e^{-J_{A_c}(T-\tau)}]^T d\tau \quad (3.24)$$

Dabei beschreibt  $J_{A_c} = \partial \dot{x}_t / \partial x_t$  die kontinuierliche Jakobimatrix für die nicht-lineare Funktion  $g_A$ . Unter Berücksichtigung der Modellfehler  $\sigma_v^2$  und  $\sigma_\omega^2$  ergibt sich für die kontinuierliche Systemkovarianz  $Q^c$  unter der Annahme, dass sich der Roboter auf einer Kreisbahn mit konstanter Geschwindigkeit befindet, folgende Matrix:

$$Q^c = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\omega^2 \end{bmatrix} \quad (3.25)$$

Für die Systemkovarianz  $Q_t$  muss auch hier wieder eine Fallunterscheidung für die beiden Fälle  $\omega \neq 0$  und  $\omega = 0$  erfolgen.

### 3.2.3 Sensormodell

Das Sensormodell beschreibt die Messübergangswahrscheinlichkeit  $p(y_t|x_t)$  mit Hilfe eines geeigneten Modells. Dieses stellt eine Beziehung zwischen der Messung  $y_t$  und dem Zustand  $x_t$  dar. Der Zusammenhang kann wie bei dem Systemmodell entweder linear (vgl. Gleichung 3.14) oder nicht-linear (vgl. Gleichung 3.17) sein.

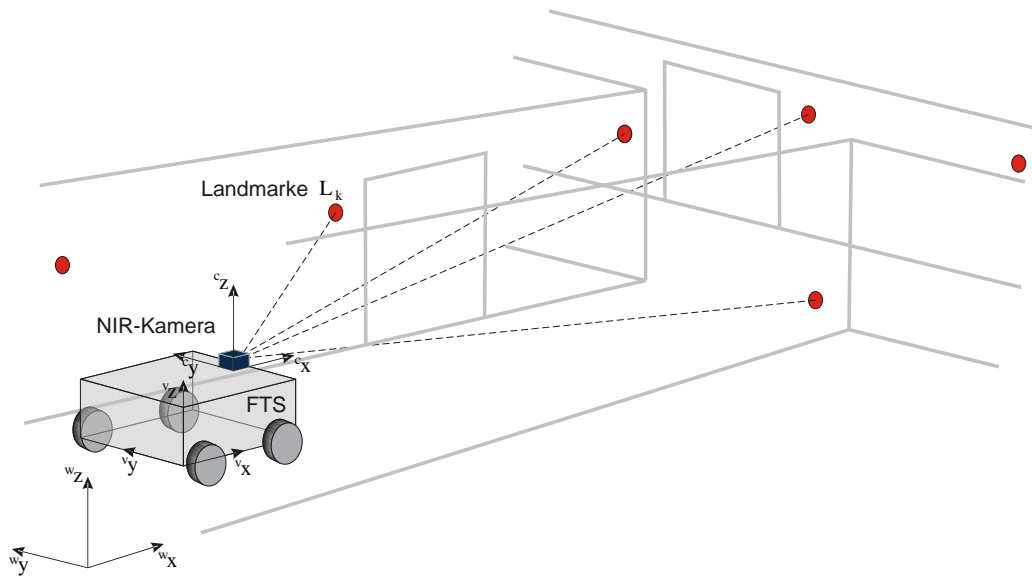
Die Messungenauigkeit wird dabei durch die Messkovarianzmatrix  $R$  abgebildet und beschreibt dabei den Einfluss der Messung auf das Schätzergebnis. Der Messvektor  $y_t$



beschreibt Messgrößen, die den Messraum abbilden, und wird wie folgt dargestellt:

$$y = [{}^i x, {}^i y]^T \quad (3.26)$$

Dabei beschreiben  ${}^i x$  und  ${}^i y$  die Bildkoordinaten einer Landmarke (vgl. Abschnitt 2.2), die durch Abbildung der Koordinaten  $L_k = [{}^w x, {}^w y, {}^w z]^T$  bezüglich eines Referenzkoordinatensystems in das Kamerabild entstanden sind. Aufgrund dieser Abbildung stellt dieses System ein nicht-lineares Modell dar und wird durch Gleichung 3.17 beschrieben.



**Abbildung 3.2:** Veranschaulichung des Sensormodells mit verschiedenen Koordinatensystemen

Für die Beschreibung dieses Modells sind einige Transformationen nötig. Ausgehend von einem Referenzsystem (Weltkoordinatensystem  ${}^w$ ), indem sich sowohl der Roboter bewegt als auch die Landmarken mit ihrer Position befinden, müssen die Landmarken in den Messraum transformiert werden. Der Messraum stellt ein sich aus den Messgrößen ergebendes System dar. Das bedeutet, der Messraum beinhaltet den Informationsgehalt (Kamerabild) einer Kamera.

Um einen Punkt  ${}^w \underline{x} = [{}^w x, {}^w y, {}^w z, 1]^T$  aus dem Weltkoordinatensystem  ${}^w$  in den Messraum zu transformieren, wird dieser als erstes in das Fahrzeugkoordinatensystem  ${}^v$  transformiert (Abbildung 3.2).

Unter Berücksichtigung des Verschiebungsvektors  ${}^w \underline{x}_v = [{}^w x_v, {}^w y_v, {}^w z_v, 1]$  und des

Rotationswinkels  ${}^w\varphi_v$  für das Fahrzeugkoordinatensystem unter der Annahme, dass der Rotationswinkel eine Rotation um die  $z$ -Achse darstellt, ergibt sich aus Abschnitt 2.3.1 folgende Gleichung in homogenen Koordinaten:

$$\begin{aligned}
 {}^v\underline{x} &= R_z \cdot T \cdot {}^w\underline{x} \\
 &= \begin{bmatrix} \cos({}^w\varphi_v) & \sin({}^w\varphi_v) & 0 & 0 \\ -\sin({}^w\varphi_v) & \cos({}^w\varphi_v) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -{}^w x_v \\ 0 & 1 & 0 & -{}^w y_v \\ 0 & 0 & 1 & -{}^w z_v \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot {}^w\underline{x} \\
 &= \begin{bmatrix} \cos({}^w\varphi_v) & \sin({}^w\varphi_v) & 0 & -\cos({}^w\varphi_v){}^w x_v - \sin({}^w\varphi_v){}^w y_v \\ -\sin({}^w\varphi_v) & \cos({}^w\varphi_v) & 0 & \sin({}^w\varphi_v){}^w x_v - \cos({}^w\varphi_v){}^w y_v \\ 0 & 0 & 1 & -{}^w z_v \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot {}^w\underline{x} \\
 &= P_{w \rightarrow v} \cdot {}^w\underline{x} \tag{3.27}
 \end{aligned}$$

Für die Projektion des Punktes  ${}^v\underline{x}$  in die Bildebene wird dieser in das Kamerakoordinatensystem  ${}^c$  transformiert und mit Hilfe der unter Abschnitt 2.3.2 beschriebenen Methoden letztlich ins Bild  ${}^i$  abgebildet (Abbildung 3.2). Aufgrund der Position und Lage der Kamera bezüglich des Fahrzeugkoordinatensystems ergibt sich mit der Position  ${}^v\underline{x}_c = [{}^v x_c, {}^v y_c, {}^v z_c, 1]$  und der allgemeinen Rotationsmatrix  $R_c$  folgende Transformation:

$$\begin{aligned}
 {}^c\underline{x} &= R_c \cdot T_c \cdot {}^v\underline{x} \\
 &= R_c \cdot \begin{bmatrix} 1 & 0 & 0 & -{}^v x_c \\ 0 & 1 & 0 & -{}^v y_c \\ 0 & 0 & 1 & -{}^v z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot {}^v\underline{x} \\
 &= P_{v \rightarrow c} \cdot {}^v\underline{x} \tag{3.28}
 \end{aligned}$$

Die sich ergebende affine Transformationsmatrix wird zu Gunsten der Übersichtlichkeit durch  $P$  mit entsprechenden Indizes ersetzt. Ersetzt man  ${}^v\underline{x}$  mit Gleichung 3.27, ergibt sich folgende Transformation um einen Punkt  ${}^w\underline{x}$  in das Kamerakoordinatensystem abzubilden:

$$\begin{aligned}
 {}^c\underline{x} &= P_{v \rightarrow c} \cdot {}^v\underline{x} \\
 &= P_{v \rightarrow c} \cdot P_{w \rightarrow v} \cdot {}^w\underline{x} \\
 &= P_{w \rightarrow c} \cdot {}^w\underline{x}
 \end{aligned} \tag{3.29}$$

Um den Punkt  ${}^c\underline{x}$  in Bildkoordinaten zu überführen, ist eine Projektion nach 2.20 in die Bildebene nötig und es ergibt sich für die normierten Kamerakoordinaten  ${}^i\underline{x}_n$ :

$${}^i\underline{x}_n = \begin{bmatrix} c_x/c_z \\ c_y/c_z \\ 1 \end{bmatrix}$$

Unter Berücksichtigung der Verzerrung und der Kalibrierungsmatrix wird der Bildpunkt  ${}^i\underline{x}$  wie folgt ermittelt:

$${}^i\underline{x} = K \cdot (d_r + d_t) \cdot {}^i\underline{x}_n \tag{3.30}$$

Wie bereits in Abschnitt 2.3.2 erläutert, wird die tangentielle Verzerrung  $d_t$  als *null* angenommen und die radiale Verzerrung  $d_r$  nur bis zur vierten Ordnung betrachtet. Somit entsprechen die Bildkoordinaten  ${}^i\underline{x}$  einer Messung  $y$ . Um diese komplexe Transformation als Sensormodell betrachten zu können, sind folgende Parameter notwendig:

- Punkt  ${}^w\underline{x}$ : Dieser Punkt entspricht der Position der Landmarken, die somit in den Messraum transformiert werden. Da die Landmarke als statisch angenommen wird, ist die Position der Landmarke somit konstant.
- Verschiebungsvektor  ${}^w\underline{x}_v$  und Rotationswinkel  ${}^w\varphi_v$  des Fahrzeugkoordinatensystems: Dieser Vektor setzt sich aus den entsprechenden Systemgrößen des Systemmodells zusammen. Somit entspricht  ${}^wx_v = {}^wx$  und  ${}^wy_v = {}^wy$  des Systemmodells. Die Koordinate  ${}^wz_v$  wird als *null* betrachtet, da sich das Fahrzeug (FTS) nur in der x-y-Ebene des Weltkoordinatensystems bewegt. Der Rotationswinkel  ${}^w\varphi_v$  entspricht dabei der Ausrichtung des Fahrzeugs  ${}^w\gamma$ .
- Verschiebungsvektor  ${}^v\underline{x}_c$  und die Rotationsmatrix  $R_c$ : Diese Parameter beschreiben die Position und die Ausrichtung der Kamera. Diese Größen werden einmalig bestimmt und sind somit konstant.

- Kameraparameter: Diese Parameter setzen sich aus den Verzerrungskoeffizienten und den intrinsischen Parametern zusammen und wurden bereits in Abschnitt 2.3.2 ausführlich erläutert und für die verwendete Kamera angegeben. Diese Parameter werden durch Kalibrierung einmalig bestimmt und sind somit auch konstant.

Durch die Linearisierung des EKF ergibt sich folgende Abhängigkeit (vgl. Gleichung 3.17 und Tabelle 3.4 Zeile 7):

$$\tilde{y}_t = g_C(\tilde{x}_t)$$

Dabei stellt  $\tilde{y}_t$  die vorhergesagte Messung dar. Die nicht-lineare Funktion  $g_C$  beschreibt die oben erklärte Transformation. Um die Linearisierung an der Stelle  $x_{t-1}$  durchzuführen, ist die Bestimmung der Jakobimatrix  $J_{C,t}$  notwendig. Diese wird wie schon bei dem Systemmodell mittels Gleichung 3.19ff berechnet. Die Messkovarianzmatrix  $R_t$  stellt dabei die Messungenauigkeit dar und ist wie folgt angegeben:

$$R_t = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}$$

Ein Sensor stellt somit eine Beschreibung einer Landmarke mit der Position  $L_k$  für das Sensormodell dar und wird als  $S_k$  bezeichnet.

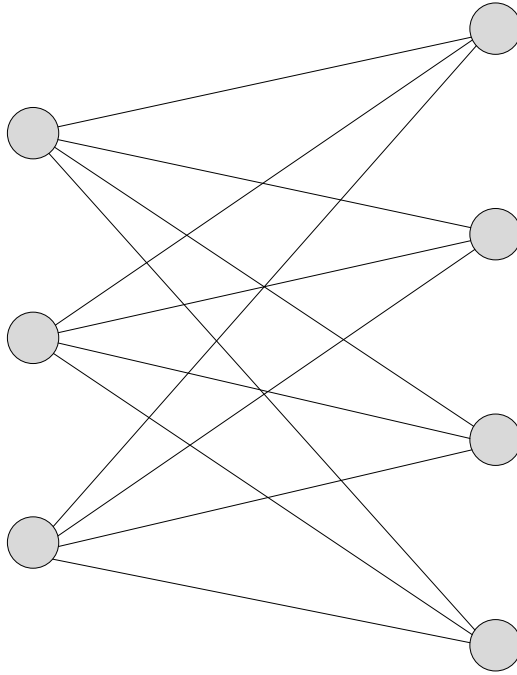
### 3.2.4 Zuordnungsmethoden

Eine Zuordnung bezeichnet hierbei die Beziehung zwischen der aktuellen Messung  $y_t$  und der vorhergesagten Messung  $\tilde{y}_t$ . Um eine Messung vorherzusagen, müssen aus einer Menge von  $N$  Landmarken die ausgewählt werden, die sich auf Grund der aktuellen Position im Sichtbereich der Kamera befindet. Diese Untermenge an vorhergesagten Messungen muss mit den tatsächlichen Messungen assoziiert werden. Das Problem dabei besteht in der Assoziation dieser Messungen. Denn auf Grund dieser Assoziation wird über den vorhergesagten Zustand  $\tilde{est}(x_t)$  (Tabelle 3.4 in Zeile 7-8) die Korrektur für den aktuellen Zustand  $est(x_t)$  durchgeführt. Das bedeutet, durch eine schlechte Assoziation der Messungen kann es dazu kommen, dass der Filter falsche Ergebnisse liefert.

Das Problem der Zuordnung entspricht dem in einem bipartiten Graph<sup>6</sup>. Ein

<sup>6</sup>Bipartite Graphen sind ein Teilgebiet der Graphentheorie.

bipartiter Graph beschreibt dabei zwei disjunkte Teilmengen, so dass zwischen den Knoten innerhalb dieser Teilmengen keine Kanten verlaufen. Abbildung 3.3 stellt ein Beispiel eines bipartiten Graphen dar.



**Abbildung 3.3:** Darstellung eines vollständigen bipartiten Graphen

Für solche Zuordnungsprobleme gibt es verschiedene Algorithmen. Die Zuordnung erfolgt über eine  $m \times n$ -Kostenmatrix. Die Kostenmatrix beschreibt dabei die Kosten zwischen  $m$  Messungen und  $n$  vorhergesagten Messungen. Die Elemente (Kosten) stellen jeweils die Entfernung einer Messung  $i$  zu einer vorhergesagten Messung  $j$  dar und werden über verschiedene Abstandsmaße (z.B. Euklidischer Abstand, City-Block-Abstand) bestimmt. Aus den Kosten ergibt sich somit eine  $m \times n$ -Matrix:

$$C = \begin{bmatrix} c_{1,1} & \cdots & c_{1,n} \\ \vdots & \ddots & \vdots \\ c_{m,1} & \cdots & c_{m,n} \end{bmatrix} \quad (3.31)$$

In dem folgenden Abschnitt werden einige für die Lösung der Aufgabe verwendeten Algorithmen kurz erläutert.

### Ungarische Methode

Die *Ungarische Methode*, auch der Algorithmus nach Munkres genannt, ist ein globales Optimierungsverfahren, welches eine  $m \times n$ -Kostenmatrix minimiert. Die Kosten werden dabei speziell für diese Aufgabe mit Hilfe des City-Block-Abstandes (auch *Manhattan-Metrik* genannt) berechnet:

$$c_{i,j}(y_{t,i}, \tilde{y}_{t,j}) = \left| x_{t,i}^y - x_{t,j}^{\tilde{y}} \right| + \left| y_{t,i}^y - y_{t,j}^{\tilde{y}} \right|$$

Dieser Algorithmus ordnet dabei die Zeilen- und Spaltenvektoren so um, dass die totalen Kosten minimal werden. Wenn die Kosten  $c_{i,j}(y_{t,i}, \tilde{y}_{t,j})$  minimal sind, entspricht die vorhergesagte Messung  $\tilde{y}_t$  gleich der tatsächlichen Messung  $y_t$ . Der Algorithmus löst das Zuordnungsproblem in Polynomialzeit mit:

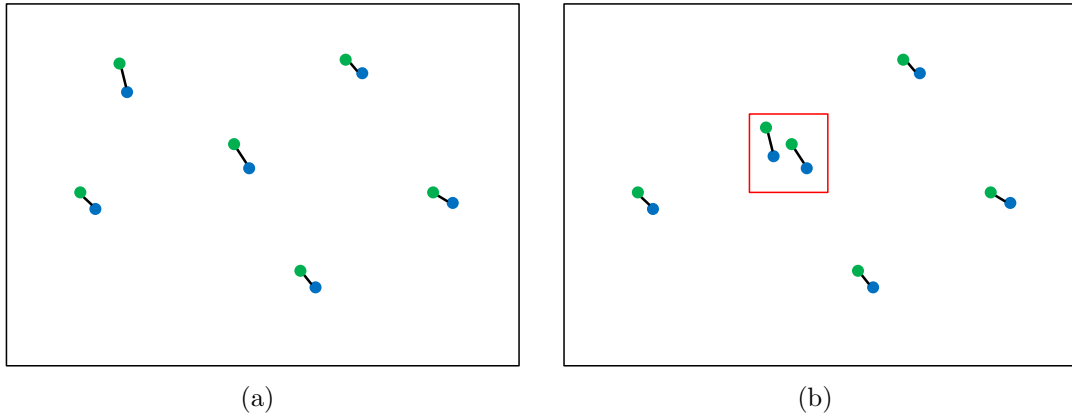
$$O(n^3)$$

In Abbildung 3.4 ist ein Beispiel für die Zuordnung mittels Ungarischer Methode dargestellt. Dabei beschreiben die blauen Punkte reale Messungen  $y_t$  und die grünen Punkte vorhergesagte Messungen  $\tilde{y}_t$ . In den meisten Fällen (Abbildung 3.4(a)) liegen vorhergesagte und tatsächliche Messung nah beieinander und können somit einfach zugeordnet werden. Jedoch kann es auch zu ungünstigen Konstellationen (Abbildung 3.4(b): Roter Kasten) kommen. Durch die globale Optimierung löst die Ungarische Methode auch hier die meisten solcher Probleme. Allerdings kann es auch mit Hilfe dieses Algorithmus zu falschen Zuordnungen kommen.

Eine genaue Erläuterung dieses Algorithmus findet sich unter [Munkres57] und [Pilgrim07].

### Nearest Neighbor

Als *Nearest-Neighbor-Methode* (Nächster-Nachbar-Methode) wird ein Optimierungsverfahren bezeichnet, welches versucht, in einer  $m \times n$ -Ähnlichkeitsmatrix maximale Ähnlichkeit zu finden. Allerdings führt dieser Algorithmus nicht zwingend zu einer optimalen Lösung. Die Ähnlichkeit wird mit Hilfe des Euklidischen Abstandes wie



**Abbildung 3.4:** Beispiel für die Ungarische Methode mit blauen Punkten als reale Messungen und grünen Punkten als vorhergesagte Messungen: (a) einfache globale Zuordnung, (b) Durch die globale Optimierung (Minimierung der Kosten) können auch Probleme bei der Zuordnung (roter Kasten) richtig gelöst werden.

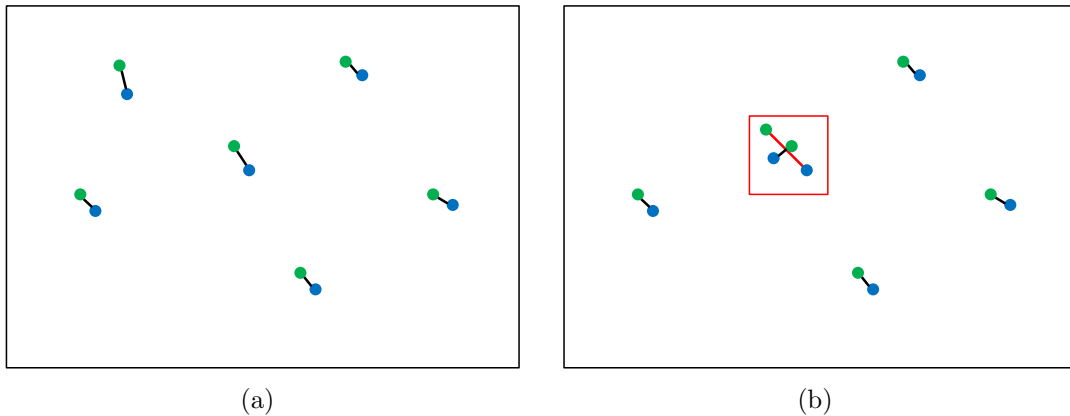
folgt beschrieben:

$$\mu_{i,j}(y_{t,i}, \tilde{y}_{t,j}) = \frac{1}{1 + \sqrt{\left(i x_{t,i}^y - i x_{t,j}^{\tilde{y}}\right)^2 + \left(i y_{t,i}^y - i y_{t,j}^{\tilde{y}}\right)^2}} \quad (3.32)$$

Mit Hilfe dieser Formel werden die Ähnlichkeit auf eins normiert, wobei 1 exakte Übereinstimmung und 0 keine Übereinstimmung darstellt. Dieser Algorithmus sucht bei jedem Durchlauf die maximale Ähnlichkeit und bestimmt somit seine Zuordnungen. Dabei werden einmal zugeordnete Elemente beim nächsten Durchlauf nicht mehr berücksichtigt. Auch hier ergibt sich für das Zuordnungsproblem die Polynomialzeit von:

$$O(n^3)$$

In Abbildung 3.5 ist beispielsweise die Zuordnung mittels *Nearest-Neighbor*-Methode dargestellt. Auch hier beschreiben die blauen Punkte reale Messungen  $y_t$  und die grünen Punkte vorhergesagte Messungen  $\tilde{y}_t$ . Dabei ordnet diese Methode in den meisten Fällen (Abbildung 3.4(a)) vorhergesagte und tatsächliche Messung richtig zu. Jedoch ergeben sich bei ungünstigen Konstellationen (Abbildung 3.4(b): Roter Kasten) falsche Zuordnungen. Dies kann somit zu einem falschen Ergebnis des Filters führen.



**Abbildung 3.5:** Beispiel für die Nearest-Neighbor-Methode mit blauen Punkten als reale Messungen und grünen Punkten als vorhergesagte Messungen: (a) einfache Zuordnung und (b) falsche Zuordnung (rote Linie) bei ungünstigen Konstellationen

### Pattern Matching

Als *Pattern Matching*<sup>7</sup> wird ein Verfahren bezeichnet, das in einem definierten Bereich das komplette Muster der vorhergesagten Messungen verschiebt und für jede Verschiebung mit Hilfe der Nearest-Neighbor-Methode die Qualität der Zuordnung bestimmt. Dabei entspricht die Qualität  $q$  gleich der Summe aller Ähnlichkeiten  $\mu_k$  der Zuordnungen aus  $m$  Messungen und  $n$  vorhergesagten Messungen.

$$q = \sum_{k=1}^{\min(m,n)} \mu_k$$

Aus dieser Menge an möglichen Zuordnungen wird die Zuordnung mit der besten Qualität ausgewählt. Dieses Verfahren hat jedoch einen entscheidenden Nachteil, dass es in der praktischen Umsetzung zuviel Zeit für die Berechnung der besten Zuordnung benötigt und kommt daher in zeitkritischen Systemen eher weniger zum Einsatz.

### 3.2.5 Erläuterungen zur Messwahrscheinlichkeit

Der Korrekturschritt (vgl. Tabelle 3.4) des Kalman-Filters erschließt sich aus der Messgleichung (Tabelle 3.1 Zeile 4). Dabei beschreibt die Messwahrscheinlichkeit (Gleichung 3.2) die Wahrscheinlichkeit einer Messung  $y_t$  unter der Bedingung, dass

<sup>7</sup>engl. für Musterzuordnung



$x_t$  schon eingetreten ist. Es ist jedoch häufig der Fall, dass zu einem Zeitpunkt  $t$  mehrere Messungen vorliegen. Somit berechnet sich die neue Schätzung  $est(x_t)$  aus einer Menge von  $k$  Messungen und dem vorhergesagten Zustand  $\widetilde{est}(x_t)$  wie folgt:

$$est(x_t) = \eta p(y_{t,1:k}|x_t) \widetilde{est}(x_t) = \eta p(y_{t,1}, \dots, y_{t,k}|x_t) \widetilde{est}(x_t)$$

Um die Korrektur des Kalman-Filters für mehrere Messungen  $y_{t,1} \dots y_{t,k}$  durchführen zu können, werden zwei Fälle unterschieden:

- Die Messungen  $y_{t,1} \dots y_{t,k}$  sind voneinander unabhängig. Beispielsweise werden bei einem System, welches Odometriedaten mittels Kalman-Filter verarbeitet, die Geschwindigkeit und die Yawrate meist unabhängig voneinander gemessen und können somit auch unabhängig voneinander betrachtet werden.
- Die Messungen  $y_{t,1} \dots y_{t,k}$  sind voneinander abhängig. Bei dem in dieser Aufgabe verwendete Messsystem sind die Daten voneinander abhängig, da alle verfügbaren Landmarken zu einem Zeitpunkt  $t$  aus der Transformation des gleichen Zustandes  $x_t$  entstanden sind und somit auch in ihrer Gesamtheit betrachtet werden müssen.

### Unabhängigkeit der Messungen

Für die Unabhängigkeit der Messungen  $y_{t,1} \dots y_{t,k}$  lässt sich die Korrektur zerlegen. Dabei gilt nach Gleichung 2.35 für die verschiedenen unabhängigen Messungen folgende Beziehung:

$$p(y_{t,1}, \dots, y_{t,k}|x_t) = p(y_{t,1}|x_t) \cdot p(y_{t,2}|x_t) \cdot \dots \cdot p(y_{t,k-1}|x_t) \cdot p(y_{t,k}|x_t) \quad (3.33)$$

Daraus ergibt sich für die Korrektur des Kalman-Filters:

$$est(x_t) = \eta p(y_{t,1:k}|x_t) \widetilde{est}(x_t) \quad (3.34)$$

$$= \eta p(y_{t,1}|x_t) \cdot p(y_{t,2}|x_t) \cdot \dots \cdot p(y_{t,k-1}|x_t) \cdot p(y_{t,k}|x_t) \widetilde{est}(x_t) \quad (3.35)$$

Da für  $p(y_{t,k}|x_t) \sim N(y_{t,k}, h(\tilde{x}_t) + H_t(x_t - \tilde{x}_t), R_t)$  und  $\tilde{est}(x_t) \sim N(x_t, \bar{x}_t, P_t)$  gilt, lässt sich die Berechnung von  $est(x_t)$  wie folgt zerlegen:

$$\begin{aligned} est_1(x_t) &= \eta_1 p(y_{t,1}|x_t) \tilde{est}(x_t) \\ est_2(x_t) &= \eta_2 p(y_{t,2}|x_t) est_1(x_t) \\ &\dots \end{aligned} \tag{3.36}$$

$$\begin{aligned} est_{k-1}(x_t) &= \eta_{k-1} p(y_{t,k-1}|x_t) est_{k-2}(x_t) \\ est(x_t) = est_k(x_t) &= \eta_k p(y_{t,k}|x_t) est_{k-1}(x_t) \end{aligned} \tag{3.37}$$

Tabelle 3.6 stellt die Berechnung des Erwartungswertes  $\hat{x}$  und der Kovarianz  $P$  für unabhängige Messungen  $y_{t,1:k}$  der neuen Schätzung  $est(x_t)$  des Erweiterten-Kalman-Filters dar.

```

1 Correction_Independent_Measurement( $\tilde{x}_t, \tilde{P}_t, y_{t,1:k}$ ):
2    $\hat{x}_{t,0} = \tilde{x}_t$ 
3    $P_{t,0} = \tilde{P}_t$ 
4
5   for all i=1:k do
6      $K_{t,i} = P_{t,i-1} J_{C,t,1}^T (J_{C,t,1} P_{t,i-1} J_{C,t,1}^T + R_{t,1})^{-1}$ 
7      $\tilde{x}_{t,i} = \hat{x}_{t,i} + K_{t,i} (y_{t,i} - g_{C,i}(\hat{x}_{t,i-1}))$ 
8      $\tilde{P}_{t,i} = (I - K_{t,i} J_{C,t,i}) P_{t,i-1}$ 
9   endfor
10
11    $\hat{x}_t = \hat{x}_{t,k}$ 
12    $P_t = P_{t,k}$ 
13   return  $\hat{x}_t, P_t$ 

```

**Tabelle 3.6:** Algorithmus der Korrektur für unabhängige Messungen  $y_{t,1:k}$

### Abhängigkeit der Messungen

Sind die Messungen  $y_{t,1} \dots y_{t,k}$  voneinander abhängig, lassen diese sich nicht mehr nach Gleichung 2.35 zerlegen. Das bedeutet, die Messwahrscheinlichkeit  $p(y_{t,1:k}|x_t)$  muss somit für alle Messungen  $y_{t,1:k}$  betrachtet werden. Da alle Messungen zu einem Zeitpunkt  $t$  zugleich verarbeitet und betrachtet werden müssen, wird ein neuer Sensor eingeführt, ein sogenannter *Supersensor*. Dieser Sensor vereint alle Messungen  $y_{t,1:k}$

und erweitert somit den Messraum. Der in dieser Aufgabe verwendete Supersensor des Erweiterten-Kalman-Filters hat folgende Gestalt:

$$\begin{aligned} y^S = y_{t,1:k} &= [y_{t,1}, y_{t,2}, \dots, y_{t,k-1}, y_{t,k}]^T \\ &= [{}^i x_1, {}^i y_1, {}^i x_2, {}^i y_2, \dots, {}^i x_{k-1}, {}^i y_{k-1}, {}^i x_k, {}^i y_k]^T \end{aligned} \quad (3.38)$$

Damit ergibt sich die Dimension von  $y^S$  aus den Dimensionen der einzelnen Messungen  $y_{t,k}$ . Die sich durch die Linearisierung ergebende Jakobimatrix  $J_{C,t}^S$  und die vorhergesagte Messung  $\tilde{y}_t$  setzen sich aus den Matrizen jeder einzelnen Messung  $y_{t,1}, \dots, y_{t,k}$  zusammen. Daraus ergibt sich der Vektor  $\tilde{y}_t^S$  für die Vorhersage der Messung:

$$\tilde{y}_t^S = \tilde{y}_{t,1:k} = \begin{bmatrix} g_{C,1}(\tilde{x}_t) \\ g_{C,2}(\tilde{x}_t) \\ \vdots \\ g_{C,k-1}(\tilde{x}_t) \\ g_{C,k}(\tilde{x}_t) \end{bmatrix} \quad (3.39)$$

Die Jakobimatrix  $J_{C,t}^S$  ist eine  $k \cdot \dim(y_t) \times \dim(x_t)$ -Matrix und hat demnach folgende Gestalt:

$$J_{C,t}^S = \begin{bmatrix} J_{C,1} \\ J_{C,2} \\ \vdots \\ J_{C,k-1} \\ J_{C,k} \end{bmatrix} \quad (3.40)$$

Die Messkovarianzmatrix  $R_t^S$  setzt sich aus den Messkovarianzen  $R_{t,1}, \dots, R_{t,k}$  zusammen. Unter der Bedingung, dass die Störungen der Messungen  $y_{t,1} \dots y_{t,k}$  nicht korreliert sind, ergibt sich:

$$R_t^S = \begin{bmatrix} \begin{bmatrix} \sigma_{x,1}^2 & 0 \\ 0 & \sigma_{y,1}^2 \end{bmatrix} & \dots & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \dots & \begin{bmatrix} \sigma_{x,k}^2 & 0 \\ 0 & \sigma_{y,k}^2 \end{bmatrix} \end{bmatrix} \quad (3.41)$$

Somit ergibt sich abschließend in Tabelle 3.7 der Korrekturschritt des Erweiterten-Kalman-Filters für abhängige Messungen  $y_{t,1:k}$ .

1	<code>Correction Dependent_Measurement</code> ( $\tilde{x}_t, \tilde{P}_t, y_{t,1:k}$ ):
2	$K_t = \tilde{P}_t (J_{C,t}^S)^T (J_{C,t}^S \tilde{P}_t (J_{C,t}^S)^T + R_t^S)^{-1}$
3	
4	$\hat{x}_t = \bar{x}_t + K_t (y_t^S - \tilde{y}_t^S)$
5	$P_t = (I - K_t J_{C,t}^S) \tilde{P}_t$
6	<code>return</code> $\hat{x}_t, P_t$

**Tabelle 3.7:** Algorithmus der Korrektur für abhängige Messungen  $y_{t,1:k}$

### 3.3 Partikel-Filter

Der Partikel-Filter ist eine nicht-parametrische Umsetzung des Bayes-Filters. Dabei wird im Gegensatz zum Kalman-Filter die A-posteriori-Wahrscheinlichkeit nicht durch Parameter beschrieben, sondern durch eine endliche Anzahl an zufälligen Zustandsstichproben, die sich auf die A-posteriori-Wahrscheinlichkeit  $est(x_t)$  beziehen.

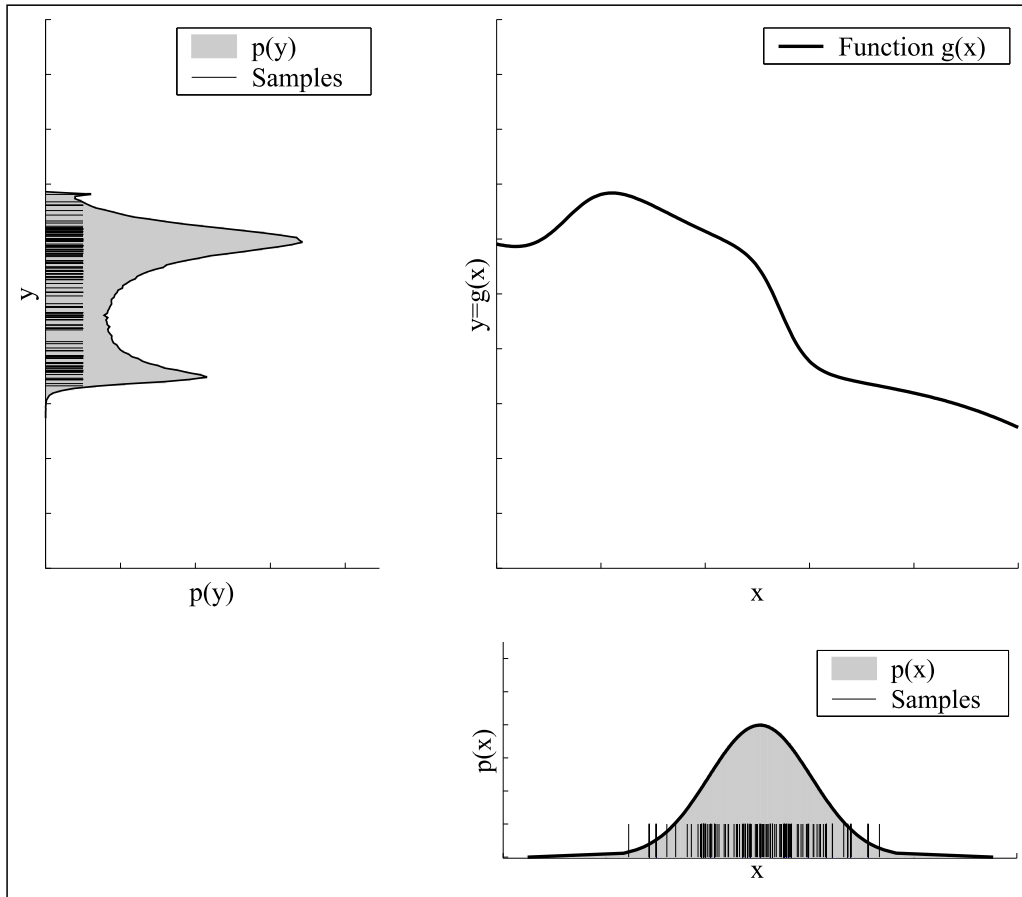
Abbildung 3.6 stellt den Grundgedanken für die Transformation einer Verteilung dar. Anstatt die Verteilung durch ihre Parameter (Exponentialfunktion als Dichtefunktion) zu beschreiben, wird diese durch eine Menge an Zustandsstichproben dieser Verteilung approximiert. Durch die Approximation einer Verteilung mit Hilfe solcher Stichproben können jegliche Verteilungen dargestellt werden.

Der Partikel-Filter bezeichnet solche Zustandsstichproben einer A-posteriori-Wahrscheinlichkeitsdichtefunktion als *Partikel*. Diese werden wie folgt bezeichnet:

$$X = X_t^{[M]} = X_t^{[1:M]} := x_t^{[1]}, x_t^{[2]}, \dots, x_t^{[M]} \quad (3.42)$$

Jedes Partikel  $x_t^{[m]}$  (mit  $1 \leq m \leq M$ ) stellt dabei eine konkrete Instanz des Zustandes  $x_t$  zum Zeitpunkt  $t$  dar. Der Partikel-Filter approximiert also die Schätzung  $est(x_t)$  durch ein Partikel-Set  $X_t^{[M]}$ . Somit stellt dieses Set die A-posteriori-Wahrscheinlichkeitsdichteverteilung  $est(x_t)$  wie folgt dar:

$$X_t^{[M]} \sim p(x_t | y_{1:t}, u_{1:t}) \quad (3.43)$$



**Abbildung 3.6:** Der untere rechte Graph stellt Stichproben einer normalverteilten Zufallsvariable  $X$  dar. Diese werden über eine nicht-lineare Funktion in die Verteilung der Zufallsvariable  $Y$  überführt [Thrun05].

Dabei gilt diese Beziehung nur asymptotisch für  $M \rightarrow \infty$ . Wie auch der Bayes-Filter bestimmt der Partikel-Filter rekursiv die Schätzung  $est(x_t)$  aus der vorherigen Schätzung  $est(x_{t-1})$ . Allerdings werden hier die Schätzungen durch Partikel-Sets repräsentiert, das heißt, dass der Partikel-Filter das Partikel-Set  $X_t^{[M]}$  rekursiv aus dem Set  $X_{t-1}^{[M]}$  berechnet.

Eine einfache Variante des Partikel-Filters ist in Tabelle 3.8 dargestellt. Dabei werden basierend auf den Stellgrößen  $u_t$  und dem vorherigen Partikel-Set  $\tilde{X}_t^{[M]}$  die vorhergesagten Partikel  $\tilde{x}_t^{[m]}$  erstellt. Die Partikel  $\tilde{x}_t^{[m]}$  werden dabei durch die Zustandsübergangsverteilung  $p(x_t|x_{t-1}, u_t)$  beschrieben. Diese zusammen ergeben die vorhergesagte Schätzung des Filters  $\tilde{est}(x_t)$ . In Zeile 6 wird für jedes Partikel der

```

1 | Algorithm Partikel-Filter ( $X_{t-1}^{[M]}, u_t, y_t$ ):
2 |    $\tilde{X}_t^{[M]} = X_t^{[M]} = \emptyset$ 
3 |
4 |   for all  $m=1:M$  do
5 |      $\tilde{x}_t^{[m]} \sim p(x_t | x_{t-1}^{[m]}, u_t)$ 
6 |      $w_t^{[m]} = p(y_t | \tilde{x}_t^{[m]})$ 
7 |      $\tilde{X}_t^{[M]} += \langle \tilde{x}_t^{[m]}, w_t^{[m]} \rangle$ 
8 |   endfor
9 |    $X_t^{[M]} = \text{Resampling}(\tilde{X}_t^{[M]})$ 
10 | return  $X_t^{[M]}$ 

```

**Tabelle 3.8:** Algorithmus des Partikel-Filters mit allgemeiner Resampling-Funktion

Wichtungsfaktor<sup>8</sup> berechnet. Diese Wichtungsfaktoren beschreiben, welchen Einfluss die Messung  $y_t$  auf das Partikel-Set hat. Somit stellt ein solcher Faktor  $w_t^{[m]}$  die Wahrscheinlichkeit dar, dass eine Messung  $y_t$  unter der Bedingung eines Partikels  $x_t^{[m]}$  auftritt, und wird wie folgt bezeichnet:

$$w_t^{[m]} = p(y_t | x_t^{[m]})$$

Die gewichteten Partikel beschreiben in gewisser Näherung die Bayes'sche A-posteriori-Wahrscheinlichkeitsverteilung  $est(x_t)$ .

Das eigentlich Besondere an dem Partikel-Filter ist in Zeile 9 dargestellt. Es werden durch einen Algorithmus  $M$  Partikel von dem Partikel-Set  $\tilde{X}_t^{[M]}$  in ein neues Partikel-Set  $X_t^{[M]}$  transformiert. Dieses Verfahren wird als *resampling* oder *importance sampling* bezeichnet. Mit Hilfe der Wichtungsfaktoren  $w_t^{[m]}$  ändert sich durch das Resampling die Verteilung der Partikel entsprechend von  $\tilde{est}(x_t)$  zu der angenäherten Verteilung  $est(x_t)$ :

$$est(x_t) = \eta p(y_t | x_t^{[m]}) \tilde{est}(x_t)$$

Für weiterführende Informationen sei hierbei unter anderem auf [Thrun05] oder [Doucet98] u. a. verwiesen.

---

<sup>8</sup>engl. *importance factor*

### 3.3.1 Systemmodell & Sensormodell

Für die Berechnung des vorhergesagten Partikel-Sets  $\tilde{X}_t^{[M]}$  ist die Kenntnis der Zustandsübergangsverteilung notwendig. Diese wird durch das Systemmodell repräsentiert. Als Systemmodell wurde, wie auch bei dem Kalman-Filter, das CTRV-Modell (Abschnitt 3.2.2) eingesetzt. Jedoch wurde für den Partikel-Filter die Dimension des Zustandsvektors  $x_t$  reduziert. Der Zustandsvektor  $x_t$  des Partikel-Filters wird im Gegensatz zu Gleichung 3.22 durch folgende Größen beschrieben:

$$x = [{}^w x, {}^w y, {}^w \gamma]^T \quad (3.44)$$

Die für das CTRV-Modell notwendige Geschwindigkeit  $v$  und Winkelgeschwindigkeit  $\omega$  wurden mit Hilfe von Stellgrößen in das Systemmodell integriert. Dabei ergibt sich für den Stellgrößenvektor  $u_t$  folgende Form:

$$u = [v, \omega]^T \quad (3.45)$$

Die Stellgrößen werden durch folgende Sensoren ermittelt:

- Geschwindigkeit  $v$  durch den Inkrementalgeber (vgl. Abschnitt 2.1.2)
- Winkelgeschwindigkeit  $\omega$  durch den Drehratensensor (vgl. Abschnitt 2.1.3)

Die Reduktion des Zustandsraumes ist notwendig, um die Anzahl an Partikeln, die für eine Approximation einer Verteilung benötigt werden, gering zu halten. Bei einer Erweiterung des Zustandsraumes steigt die Anzahl der Partikel exponentiell an. Durch die Einführung des Stellgrößenvektors  $u_t$  ergibt sich nun für die Partikel nach Gleichung 3.23 des CTRV-Modells:

$$\begin{aligned} p(x_t|x_{t-1}, u_t) &= g_A(x_{t-1}, u_t) + w_t \\ &= \begin{bmatrix} \frac{u_v}{u_\omega} [\sin(u_\omega T + {}^w \gamma_{t-1}) - \sin({}^w \gamma_{t-1})] + {}^w x_{t-1} \\ \frac{u_v}{u_\omega} [-\cos(u_\omega T + {}^w \gamma_{t-1}) + \cos({}^w \gamma_{t-1})] + {}^w y_{t-1} \\ u_\omega T + {}^w \gamma_{t-1} \end{bmatrix} + w_t \end{aligned} \quad (3.46)$$

Das Prozessrauschen  $w_t$  kann dabei entweder durch ein zufälliges Rauschen, welches die Verteilung der Schätzung besser modelliert, oder ein Systemmodell-bedingtes Rauschen repräsentiert werden.

Das Sensormodell entspricht dem in Abschnitt 3.2.3 beschriebenen Modell des Kalman-Filters. Dabei werden durch Transformationen Landmarken in das Kamerabild abgebildet.

### 3.3.2 Wichtungsfaktor und Normalisierung

Der Wichtungsfaktor  $w_t^{[m]}$  spielt für das Resampling eine wichtige Rolle. Dabei beschreibt der Wichtungsfaktor eines Partikels in wie weit die aktuelle Messung  $y_t$  den Zustand dieses Partikels wiedergibt. Die gewichteten Partikel  $x_t^{[m]}$  stellen somit in Näherung die A-posteriori-Wahrscheinlichkeit dar.

Es gibt verschiedene Möglichkeiten die Wichtungen zu bestimmen. Für diese Aufgabe wird die Berechnung der Wichtungen  $w_t^{[m]}$  über die Dichtefunktion der Messungen und der vorhergesagten Messungen vorgenommen. Dabei wird jede Messung als Normalverteilung um den Erwartungswert  $y_{k,t}$  und der Kovarianzmatrix  $R_t$  betrachtet. Der Wichtungsfaktor  $w_t^{[m]}$  eines Partikel ergibt sich dann aus der Summe aller Wahrscheinlichkeiten der vorhergesagten Messungen  $\tilde{y}_{k,t}$  entsprechend dem Mahalanobis<sup>9</sup>-Distanzmaß aus der  $n$ -dimensionalen Normalverteilung (vgl. Gleichung 2.25) zu:

$$w_t^{[m]} = \kappa \sum_{i=0}^K \sum_{j=0}^L \frac{1}{\sqrt{|2\pi R|}} \cdot e^{-\frac{1}{2}(\tilde{y}_{j,t}-y_{i,t})^T \cdot R^{-1} \cdot (\tilde{y}_{j,t}-y_{i,t})} \quad (3.47)$$

Dabei stellen  $L$  die Anzahl der vorhergesagten und  $K$  die Anzahl der tatsächlichen Messungen dar. Der Normierungsfaktor  $\kappa$  ergibt sich aus der maximal erreichbaren Wahrscheinlichkeit aller Messungen  $y_{k,t}$ :

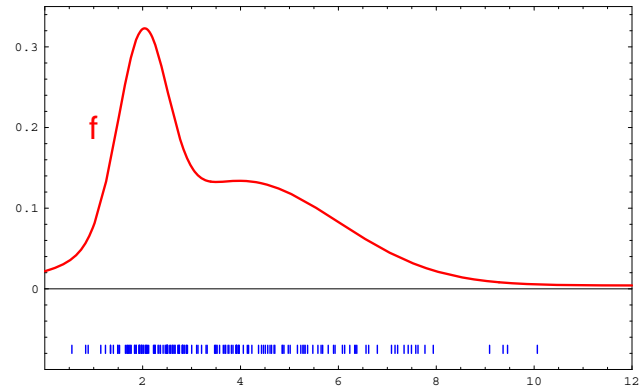
$$\kappa^{-1} = \sum_{i=0}^K \sum_{j=0}^K \frac{1}{\sqrt{|2\pi R|}} \cdot e^{-\frac{1}{2}(y_{j,t}-y_{i,t})^T \cdot R^{-1} \cdot (y_{j,t}-y_{i,t})} \quad (3.48)$$

Um den Rechenaufwand gering zu halten, wird die  $n$ -dimensionale durch eine 1-dimensionale Normalverteilung und die Kovarianzmatrix  $R_t$  mit  $\sigma_x^2 = \sigma_y^2 = \sigma_R^2$  ersetzt.

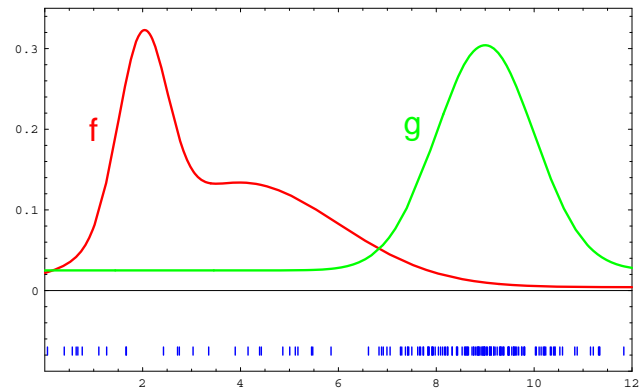
Um die Wichtungsfaktoren genauer erläutern zu können, sind einige Betrachtungen notwendig. Der Erwartungswert einer Wahrscheinlichkeitsdichtefunktion  $f$  kann durch den einer anderen Dichtefunktion  $g$  ausgedrückt werden. Der Erwartungswert einer

<sup>9</sup>Als Mahalanobis-Distanz bezeichnet man ein Distanzmaß zwischen Punkten in einem mehrdimensionalen Vektorraum.

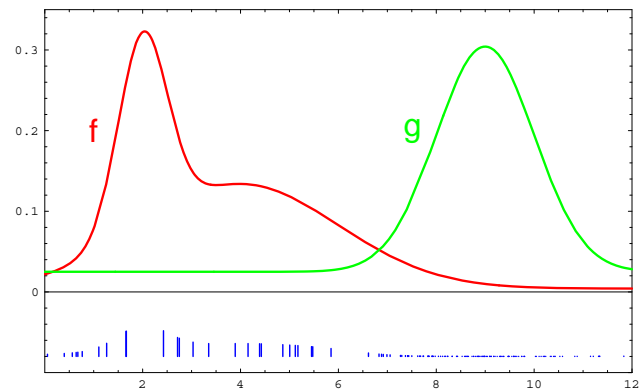




(a)



(b)



(c)

**Abbildung 3.7:** Veranschaulichung der Wichtungsfaktoren des Partikel-Filters: (a) Gesucht ist die Approximation der unbekanntes Zieldichtefunktion  $f$ . (b) Es können jedoch nur Zustandsstichproben der bekannten Dichtefunktion  $g$  benutzt werden. (c) Ein Partikel von  $f$  wird durch den Wichtungsfaktor  $f(x)/g(x)$  eines jeden Partikels  $x$  bestimmt [Thrun05].

Teilmenge  $A$  einer Zufallsvariablen  $X$  ergibt sich dann wie folgt:

$$\begin{aligned}
 E_f [I(x \in A)] &= \int f(x) I(x \in A) dx \\
 &= \int \underbrace{\frac{f(x)}{g(x)}}_{:=w(x)} g(x) I(x \in A) dx \\
 &= E_g [w(x) I(x \in A)]
 \end{aligned} \tag{3.49}$$

Dabei stellt  $I$  eine Indikatorfunktion dar und wird folgendermaßen beschrieben:

$$I(x \in A) = \begin{cases} 1, & \text{falls } x \in A \\ 0, & \text{sonst} \end{cases}$$

Als Wichtungsfaktor wird hier  $w(x)$  bezeichnet, welcher den Unterschied zwischen  $f$  und  $g$  beschreibt. Als Voraussetzung der Gleichung muss  $f(x) > 0 \rightarrow g(x) > 0$  gelten. Der Partikel-Filter macht sich diese Betrachtungen zu Nutze. Abbildung 3.7(a) stellt die Dichtefunktion  $f$ , welche auch *target distribution* genannt wird, dar. Da Partikel nicht direkt von der unbekanntem Dichtefunktion  $f$  abgeleitet werden können, werden sie von einer bekannten Dichtefunktion  $g$ , auch als *proposal distribution* bezeichnet, erstellt (Abbildung 3.7(b)). Das resultierende Partikel-Set ist im unteren Teil der Abbildung 3.7(b) dargestellt und entsprechend der Funktion  $g$  verteilt. Für eine Teilmenge  $A$  der Zufallsvariablen  $X$  konvergiert eine empirische Anzahl an Partikeln, die in  $A$  liegen, zu dem Integral der Funktion  $g$  über  $A$  zu:

$$\frac{1}{M} \sum_{m=1}^M I(x^{[m]} \in A) \rightarrow \int_A g(x) dx$$

Um die Differenz zwischen den beiden Funktionen  $f$  und  $g$  auszugleichen werden die Partikel  $x^{[m]}$  entsprechend gewichtet:

$$w^{[m]} = \frac{f(x^{[m]})}{g(x^{[m]})} \tag{3.50}$$

Diese Wichtungsfaktoren sind in Abbildung 3.7(c) dargestellt. Die vertikalen Balken entsprechen den Wichtungsfaktoren der Partikel. Aus den Wichtungsfaktoren

entsprechend 3.49 ergibt sich nun:

$$\left[ \sum_{m=1}^M w^{[m]} \right]^{-1} \sum_{m=1}^M I(x^{[m]} \in A) w^{[m]} \rightarrow \int_A f(x) dx$$

Mit anderen Worten konvergieren die gewichteten Partikel, die aus der Dichtefunktion  $g$  erstellt wurden, gegen die gesuchte Dichtefunktion  $f$ . Ausführlichere Informationen bezüglich der Wichtung von Partikel ist in [Thrun05] zu finden. Für den Partikel-Filter stellt die Dichtefunktion  $f$  die gesuchte Schätzung  $est(x_t)$  dar. Unter der Annahme, dass das Partikel-Set  $X_{t-1}^{[M]}$  entsprechend der vorherigen Schätzung  $est(x_{t-1})$  verteilt ist, beschreibt die Dichtefunktion  $g$  den vorhergesagten Zustand  $\widetilde{est}(x_t)$ .

### 3.3.3 Mathematische Betrachtungen des Partikel-Filters

Um den Partikel-Filter mathematisch beschreiben zu können, ist es notwendig ein Partikel als Zustandsstichprobe aller Zustandssequenzen zu betrachten.

$$x_{0:t}^{[m]} = x_0^{[m]}, x_1^{[m]}, \dots, x_t^{[m]}$$

Dieser Partikel-Filter berechnet die A-Posteriori-Wahrscheinlichkeitsverteilung über alle Zustandssequenzen

$$est(x_{0:t}) = p(x_{0:t} | u_{1:t}, y_{1:t})$$

anstatt der Schätzung  $est(x_{0:t}) = p(x_t | u_t, y_t)$ . Diese Definition dient lediglich der Herleitung des Filters in Tabelle 3.8. Die Wahrscheinlichkeitsverteilung  $est(x_{0:t})$  erhält man aus der Herleitung in Abschnitt 3.1 zu:

$$\begin{aligned} p(x_{0:t} | u_{1:t}, y_{1:t}) &\stackrel{\text{Bayes}}{=} \eta p(y_t | x_{0:t}, y_{1:t-1}, u_{1:t}) p(x_{0:t} | u_{1:t}, y_{1:t-1}) \\ &\stackrel{\text{Markov}}{=} \eta p(y_t | x_t) p(x_{0:t} | u_{1:t}, y_{1:t-1}) \\ &= \eta p(y_t | x_t) p(x_t | x_{0:t-1}, u_{1:t}, y_{1:t-1}) p(x_{0:t-1} | u_{1:t}, y_{1:t-1}) \\ &\stackrel{\text{Markov}}{=} \eta p(y_t | x_t) p(x_t | x_{t-1}, u_t) p(x_{0:t-1} | u_{1:t-1}, y_{1:t-1}) \end{aligned} \quad (3.51)$$

Durch das Betrachten aller Zustände in der A-posteriori-Wahrscheinlichkeit kann das Integral ersetzt werden. Wenn das Partikel-Set  $X_{0:t-1}^{[M]}$  entsprechend der vorherigen Schätzung  $est(x_{0:t-1})$  verteilt ist, so wird das  $m$ -te Partikel  $x_t^{[m]}$  (Tabelle 3.8 Zeile 5)

von der vorhergesagten Schätzung  $\widetilde{est}(x_{0:t})$  erstellt:

$$p(x_t|x_{t-1}, u_t) est(x_{0:t-1}) = p(x_t|x_{t-1}, u_t) p(x_{0:t-1}|u_{1:t-1}, y_{1:t-1})$$

Dieser Schritt entspricht dabei der Vorhersage des Bayes-Filters (Tabelle 3.1 Zeile 3). Entsprechend Gleichung 3.50

$$\begin{aligned} w_t^{[m]} &= \frac{\text{target distribution}}{\text{proposal distribution}} \\ &= \frac{\eta p(y_t|x_t) p(x_t|x_{t-1}, u_t) p(x_{0:t-1}|u_{1:t-1}, y_{1:t-1})}{p(x_t|x_{t-1}, u_t) p(x_{0:t-1}|u_{1:t-1}, y_{1:t-1})} \\ &= \eta p(y_t|x_t) \end{aligned} \tag{3.52}$$

und dem Resampling der Partikel unter Berücksichtigung ihrer Wichtungsfaktoren  $w_t^{[m]}$  ergibt sich eine Verteilung, die entsprechend der vorhergesagten Schätzung  $\widetilde{est}(x_{0:t})$  verteilt ist:

$$\eta w_t^{[m]} p(x_t|x_{t-1}, u_t) p(x_{0:t-1}|u_{1:t-1}, y_{1:t-1}) = est(x_{0:t}) \tag{3.53}$$

Die Konstante  $\eta$  unterscheidet sich hierbei von der in Gleichung 3.52. Durch Betrachtung der Partikel  $x_{0:t}^{[m]}$  für alle Zustände der Schätzung  $est(x_{0:t})$  sind die Partikel  $x_t^{[m]}$  auch entsprechend der Schätzung  $est(x_t)$  verteilt. Durch diese Annahme folgt der in Tabelle 3.8 erläuterte Algorithmus. Diese Herleitung gilt jedoch nur für  $M \rightarrow \infty$ . Für eine ausführliche Darstellung dieser Thematik sei auf [Thrun05] verwiesen.

### 3.3.4 Resampling

Mit Hilfe des *Resampling*-Algorithmus werden Partikel zu der Schätzung  $est(x_t)$  zusammengeführt. Dabei wird über die Wichtungsfaktoren  $w_t^{[m]}$  die vorhergesagte Schätzung  $\widetilde{est}(x_t)$ , repräsentiert durch das Partikel-Set  $\widetilde{X}_t^{[M]}$ , in die neue Schätzung  $est(x_t)$ , repräsentiert durch das Partikel-Set  $X_t^{[M]}$ , transformiert. Partikel mit einem hohen Wichtungsfaktor (Wahrscheinlichkeit) werden somit häufig für die neue Schätzung  $est(x_t)$  ausgewählt und Partikel mit niedrigem Wichtungsfaktor sind nicht mehr in dem Partikel-Set  $X_t^{[M]}$  enthalten. Der Resampling-Schritt ist eine Implementierung des Darwin'schen Gesetzes „Der Stärkere überlebt“.

Es gibt verschiedene Möglichkeiten das Resampling durchzuführen. Die hier ver-

wendete Variante wird als „*low variance sampling*“ bezeichnet. Anstatt voneinander unabhängige Partikel auszuwählen, werden diese durch einen weiteren sequentiellen Zufallsprozess ermittelt.

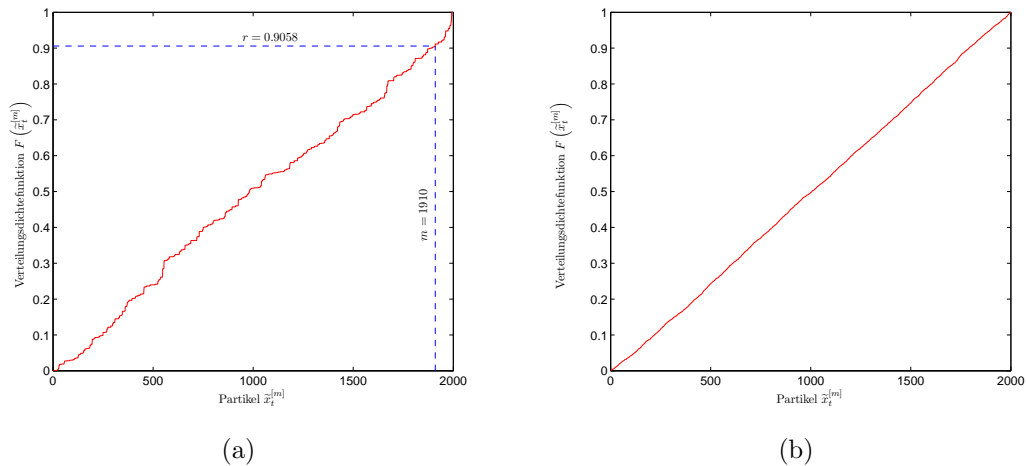
```

1 | Algorithm Low_Variance_Resampling( $\tilde{X}_t^{[M]}$ ):
2 |
3 |   for all m=1:M do
4 |      $r = rand(0; 1)$ 
5 |      $\tilde{x}_t^{[m]} = \text{Binary\_Search}(r, \tilde{X}_t^{[M]})$ 
6 |     add  $\tilde{x}_t^{[m]}$  to  $X_t^{[M]}$ 
7 |   endfor
8 |
9 | return  $X_t^{[M]}$ 

```

**Tabelle 3.9:** Low Variance Resampling Algorithmus des Partikel-Filters

Tabelle 3.9 stellt eine Implementierung des Low-Variance-Resamplings dar. Dabei wird durch einen sequentiellen Zufallsprozess ein Partikel entsprechend seinem Wichtungsfaktor ausgewählt. Die Auswahl erfolgt dabei über eine binäre Suche der Verteilungsdichtefunktion des Partikel-Sets  $\tilde{X}_t^{[M]}$ .



**Abbildung 3.8:** Veranschaulichung der Verteilungsdichtefunktion  $F(\tilde{x}_t^{[m]})$  für den Partikel-Filter mit 2000 Partikeln: (a) Verteilungsdichtefunktion kurz nach der Initialisierung, (b) Verteilungsdichtefunktion nach erfolgter Initialisierung

In Zeile 4 wird ein Zufallsprozess gestartet und eine Zufallszahl  $r$  im Bereich  $[0; 1]$  ausgewählt. Mit Hilfe dieser Zufallszahl  $r$  wird nun das entsprechende Partikel  $\tilde{x}_t^{[m]}$  mit dem Index  $m$  gesucht. Abbildung 3.8 veranschaulicht diesen Zufallsprozess. Die Zufallszahl  $r$  entspricht der kumulierten Wahrscheinlichkeit der Verteilungsdichtefunktion  $F(\tilde{x}_t^{[m]})$ . Beispielhaft ist in Abbildung 3.8(a) für eine Zufallszahl  $r$  der dazugehörige Index  $m$  abgebildet. Die binäre Suche (Zeile 5) ermittelt durch ständiges Halbieren des Partikel-Sets  $\tilde{X}_t^{[M]}$  den entsprechenden Index  $m$ . Voraussetzung für diese Suche sind sortierte Elemente, die durch die Verteilungsdichtefunktion  $F(\tilde{x}_t^{[m]})$  gegeben sind. Zuerst wird das Partikel-Set  $\tilde{X}_t^{[M]}$  halbiert und das Partikel

$$\tilde{x}_t^{[m=M/2]}$$

mit einer Zufallszahl  $r$  verglichen. Danach ergibt sich die Untermenge  $U_t^{\tilde{X}}$  des Partikel-Sets  $\tilde{X}_t^{[M]}$  wie folgt:

$$U_t^{\tilde{X}} = \begin{cases} \tilde{X}_t^{[1:M/2]}, & \text{wenn } w_t^{[m=M/2]} \geq r \\ \tilde{X}_t^{[M/2:M]}, & \text{wenn } w_t^{[m=M/2]} < r \end{cases}$$

Die Untermenge wird entsprechend so lange halbiert, bis schließlich nur ein Partikel übrig bleibt. Somit wird für jede Zufallszahl  $r$  das entsprechende Partikel  $\tilde{x}_t^{[m]}$  ermittelt. Die Komplexität der binären Suche ist durch

$$O(\log M)$$

gegeben. Dadurch ergibt sich für  $M$  Partikel die Komplexität des Low-Variance-Resamplings zu:

$$O(M \log M) \tag{3.54}$$

Diese effiziente Implementierung des Resampling-Algorithmus verbessert die Performance des Partikel-Filters. Die Laufzeit ist von besonderer Bedeutung bei der Benutzung des Partikel-Filters, da aufgrund der großen Anzahl an Partikeln viele Berechnungen durchgeführt werden müssen. Für weitere Informationen bezüglich Resampling-Methoden sei auf [Doucet98] und [Thrun05] verwiesen.

### 3.3.5 Bestimmung des Schätzergebnisses

Für die eigentliche Bestimmung des Schätzergebnisses des Partikel-Filters gibt es verschiedene Möglichkeiten. Dabei repräsentiert das Schätzergebnis die bestmögliche Aussage über die Schätzung  $est(x_t)$ . Für den Partikel-Filter kann das Schätzergebnis beispielsweise wie folgt berechnet werden:

- Erwartungswert der Schätzung  $est(x_t)$ :

$$\hat{x}_t = \frac{1}{M} \sum_{m=1}^M w_t^{[m]} x_t^{[m]}$$

- Partikel mit dem größten Wichtungsfaktor  $w_t^{[m]}$

$$\hat{x}_t = x_t^{[m=\max(w_t^{[m]})]}$$

## 3.4 Vergleich Partikel-Filter und Kalman-Filter

In diesem Abschnitt werden die Unterschiede zwischen dem Kalman- und dem Partikel-Filter dargestellt.

- **SCHÄTZUNG**  $est(x_t)$ :

Der EKF benutzt kontinuierliche Zustandsvariablen  $x_t$  und bestimmt seine Schätzung durch die Gauß'sche Normalverteilung. Wichtige Parameter für diese Verteilung sind der Erwartungswert  $\hat{x}_t$  und die Kovarianz  $P_t$ .

Der Partikel-Filter ist an keine spezifische Verteilung gebunden. Der Filter approximiert seine Schätzung durch eine finite Anzahl an Partikeln, repräsentiert durch ein Partikel-Set  $X_t^{[M]}$ . Dabei beschreiben Partikel  $x_t^{[m]}$  einen möglichen Zustand einer Verteilung.

- **ZUSTANDSÜBERGANGSGLEICHUNG**:

Der EKF beschreibt die Zustandsübergangsgleichung über nicht-lineare Zusammenhänge des vorhergesagten und des vergangenen Zustandes. Da der EKF auf einer Normalverteilung beruht, werden die nicht-linearen Zusammenhänge

approximiert (vgl. Abschnitt 3.2.1). Die Gleichung, gemäß 3.16, lautet wie folgt:

$$x_t = g_A(x_{t-1}, u_t) + w_t$$

Der Partikel-Filter ermittelt die vorhergesagte Schätzung  $\widetilde{est}(x_t)$  über die Zustandsübergangsverteilung  $p(x_t|u_t, x_{t-1}^{[m]})$ . Dabei ergibt sich für jedes vorhergesagte Partikel entsprechend Tabelle 3.8 Zeile 5:

$$\widetilde{x}_t^{[m]} \sim p(x_t|u_t, x_{t-1}^{[m]})$$

- **MESSGLEICHUNG:**

Der EKF stellt auch die Messungen über nicht-lineare Systeme, die aufgrund der Voraussetzung einer Normalverteilung approximiert werden müssen, dar. Somit ergibt sich als Messgleichung des EKF (Gleichung 3.17):

$$y_t = g_C(x_t) + \nu_t$$

Der Partikel-Filter stellt die A-posteriori-Wahrscheinlichkeitsverteilung  $est(x_t)$  über seine Wichtungsfaktoren  $w_t^{[m]}$  dar. Diese beschreiben, wie gut die aktuelle Messung  $y_t$  den Zustand eines Partikel  $x_t^{[m]}$  wiedergibt. In gewisser Näherung beschreiben diese gewichteten Partikel die Schätzung  $est(x_t)$ .

- **Globale Initialisierung:**

Der EKF ist nicht in Lage für die beschriebene Aufgabe eine Initialisierung durchzuführen, obwohl dieser, gemäß Abschnitt 3.2, einen Initialzustand benötigt.

Mit Hilfe des Partikel-Filters ist es möglich eine Initialisierung durchzuführen. Dabei werden Partikel willkürlich im Zustandsraum verteilt. Diese Partikel konvergieren nach einiger Zeit meistens zu dem eigentlichen Zustand. Die Initialisierung dieses Filters wird im Abschnitt 4.1 näher erläutert.

- **RECHENAUFWAND:**

Für den EKF existieren sehr schnelle Algorithmen. Da dieser Filter nur wenige Parameter besitzt, die berechnet werden müssen, ist die Laufzeit relativ gering. Der Partikel-Filter besitzt aufgrund seiner häufig hohen Anzahl an Partikeln



einen deutlich größeren Rechenaufwand. Diesen kann man durch geschickte Algorithmen zwar auch minimieren, allerdings sind im Vergleich zum EKF deutlich mehr Ressourcen notwendig.

## 4 Ergebnisse

In diesem Kapitel werden verschiedene Eigenschaften des Partikel-Filters und des Kalman-Filters untersucht und verglichen. Desweiteren werden Ergebnisse eines Test-szenarios ausgewertet. Zu den untersuchten Eigenschaften gehören:

- INITIALISIERUNG

Die Initialisierung ist ein wichtiger Bestandteil der Filter, da durch diese der Initialzustand  $est(x_0)$  bestimmt wird. In diesem Abschnitt wird sich damit auseinandergesetzt.

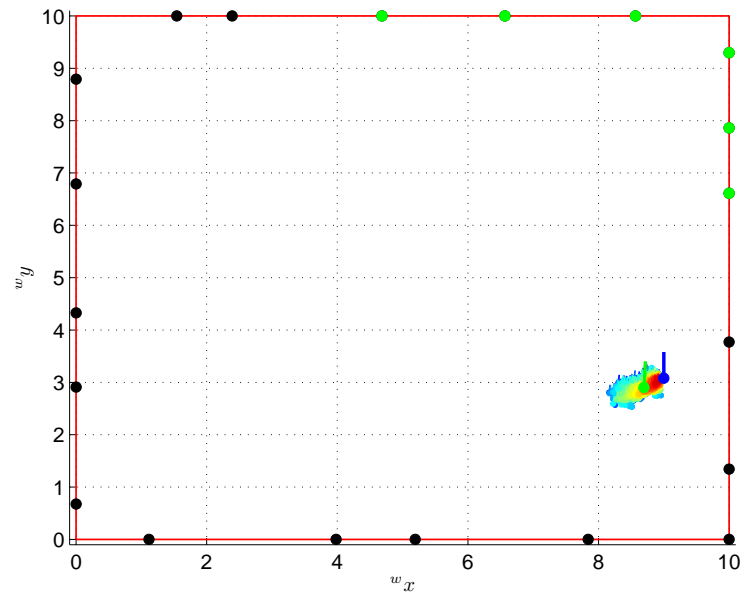
- BEWEGUNGSMODELL

Dieser Abschnitt beschreibt den Einfluss des Bewegungsmodells auf die Filter und die dabei entstehenden Ergebnisse.

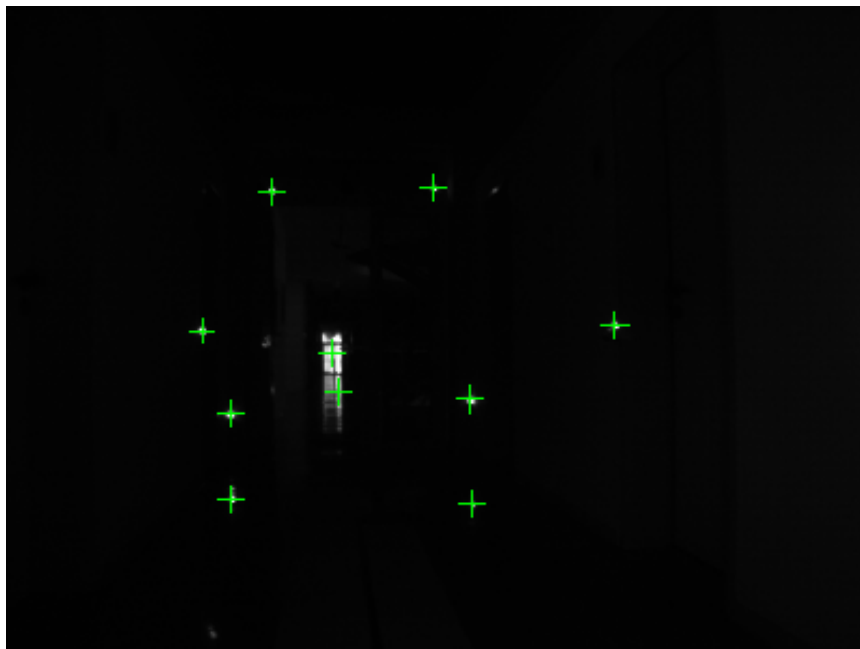
- TESTSZENARIO

Mit Hilfe einer Testfahrt werden reale Ergebnisse ausgewertet und für beide Filter verglichen.

Da die Abbildungen der Ergebnisse einheitlich sind, werden hier einige Merkmale erklärt. Abbildung 4.1 stellt exemplarisch eine Darstellung der Positionsbestimmung mittels Partikel-Filter dar. Dabei beschreiben die schwarzen/grünen Punkte am Rand in Abbildung 4.1(a) Landmarken, wobei die sich im Sichtbereich befindlichen Landmarken grün markiert werden. Der große blaue Punkt drückt die eigentliche Position und die Richtung (dünner blauer Strich) aus. Die Schätzung wird durch einen großen grünen Punkt und einen dünne Linie für die entsprechende Richtung dargestellt. Alle Partikel werden durch kleine Punkte mit der entsprechenden Richtung beschrieben. Die Farbe (blau - rot) gibt dabei die Wahrscheinlichkeit (Wichtungsfaktor) dieses Partikels an. Abbildung 4.1(b) stellt die Aufnahme einer Kamera dar, aus welcher anschließend durch einen Algorithmus die Landmarken (grüne Kreuze) extrahiert werden.



(a)



(b)

Abbildung 4.1: Beispiel für die Darstellung des Partikel-Filters

## 4.1 Initialisierung

Für die Bestimmung des Initialzustandes  $est(x_0)$  wird nur der Partikel-Filter betrachtet. Der Erweiterte-Kalman-Filter ist nicht in der Lage eine Initialisierung durchzuführen. Dies begründet sich aus der Tatsache, dass für den in dieser Aufgabe verwendeten EKF-Algorithmus die Kenntnis eines Initialzustandes  $est(x_0)$  Voraussetzung ist. Mit anderen Worten ist es dem EKF nicht möglich Messungen zuzuordnen ohne die Kenntnis eines Initialzustandes. Aus diesem Grund muss man den EKF nahe der „wahren“ Schätzung initialisieren, damit dieser korrekt arbeiten kann. Durch die Messkovarianzen der Messungen  $y_t$  wird das Einschwingverhalten des Filters bestimmt, das heißt, wie schnell sich dieser der eigentlichen Schätzung annähert.

Im Gegensatz zum EKF ist es dem Partikel-Filter möglich seinen Initialzustand  $est(x_0)$  zu bestimmen. Es gibt verschiedene Methoden einen Partikel-Filter zu initialisieren. Die für diese Aufgabe untersuchten Initialisierungsmethoden sind:

- Zufällige Verteilung der Partikel
- Verteilung der Partikel über ein Grid<sup>1</sup>

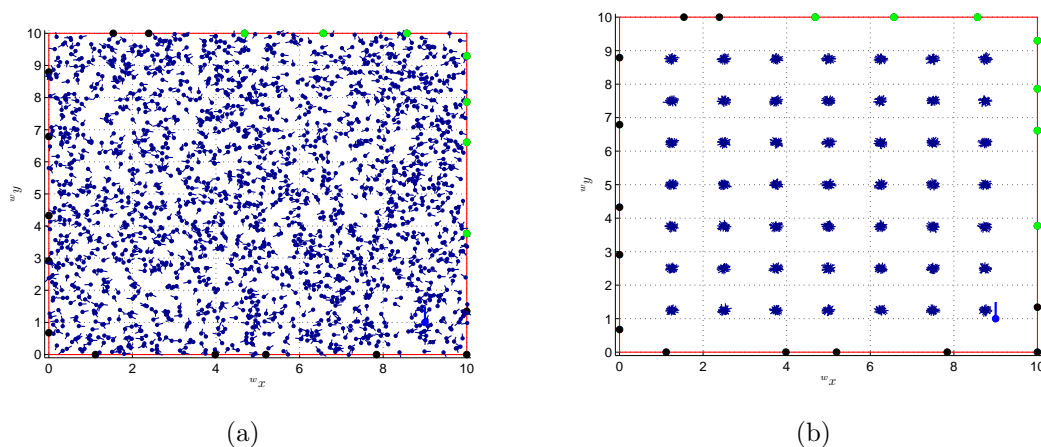
Dabei werden die Partikel zum Zeitpunkt  $t = 0$  entsprechend einer dieser Methoden im Zustandsraum verteilt. Abbildung 4.2 stellt diese zwei Methoden für 2000 Partikel in einem  $10m \times 10m$  großen Raum dar. Die blauen Punkte entsprechen dabei jeweils einem Partikel und die dünnen Striche jedes Partikels sollen exemplarisch die Richtung dieses Partikels darstellen. Der Unterschied zwischen diesen Methoden ist dabei deutlich sichtbar.

Bei einer zufälligen Verteilung werden die Partikel über dem Zustandsraum gleichverteilt (Abbildung 4.2(a)). Es ist somit reiner Zufall, wenn Partikel sich im Bereich der „wahren“ Position befinden. Für die Initialisierung mittels Grid werden die Partikel an Gitterpunkten positioniert und in alle Richtungen ausgerichtet. Abbildung 4.2(b) stellt eine solche Initialisierung dar. Dabei ist gut zu erkennen, dass sich hier die Partikel nur an bestimmten Positionen befinden und in alle Richtungen verteilt sind.

Beide Methoden unterscheiden sich im Wesentlichen nur durch ihre Verteilung und führen meist zu einer erfolgreichen Initialisierung. Diese hängt jedoch hauptsächlich von zwei weiteren Größen ab:

---

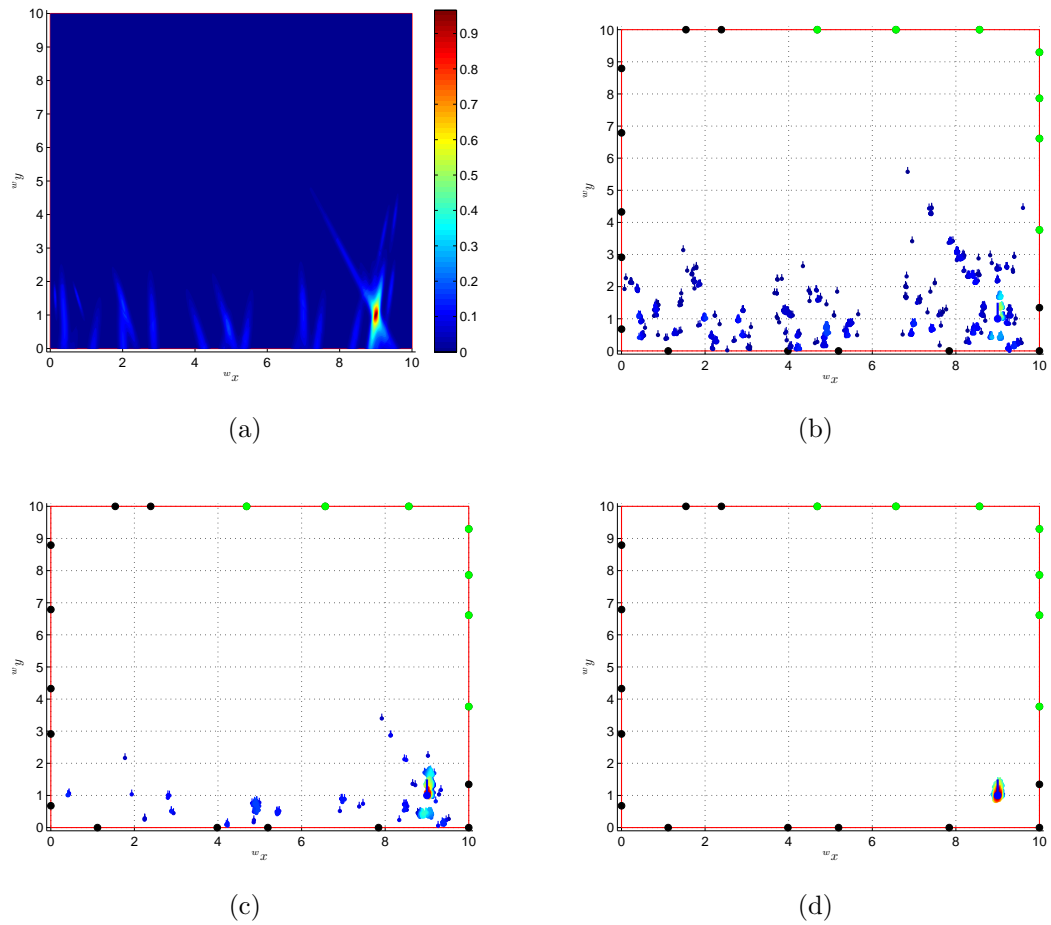
<sup>1</sup>engl. für Gitter



**Abbildung 4.2:** Initialisierungsmethoden des Partikel-Filters mit 2000 Partikeln nach dem Prädiktionsschritt in einem  $10m \times 10m$  großen Raum: (a) Initialisierung mittels zufälliger Verteilung der Partikel und (b) Initialisierung der Partikel über ein Grid

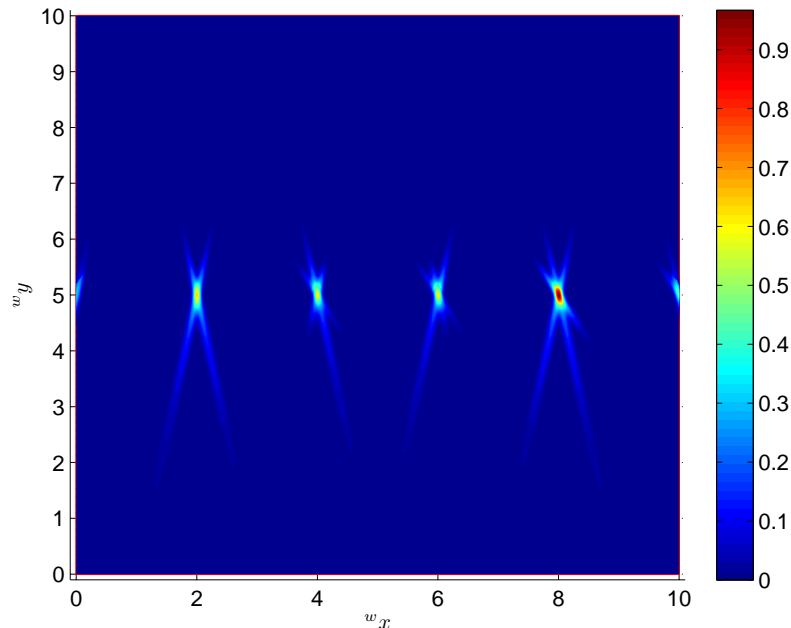
- Anzahl der Partikel
- Größe des zu initialisierenden Zustandsraumes

Je größer die Anzahl der Partikel desto dichter ist der Zustandsraum verteilt und es wird wahrscheinlicher, dass sich ein Partikel an der gesuchten Position befindet. Jedoch steigt dabei der Rechenaufwand stark an. Ein zweiter Faktor ist natürlich die Größe des Zustandsraumes. Das bedeutet einerseits je mehr Dimensionen der Zustandsvektor hat und andererseits je größer der Initialisierungsbereich des Zustandsraumes ist, desto mehr Partikel sind notwendig um eine vernünftige Initialisierung zu erreichen. Abbildung 4.3 zeigt beispielsweise die Initialisierung eines Partikel-Filters mit 1000 Partikeln, einem  $\sigma_R = 5px$  und einem konstanten  $w_\gamma = 90^\circ$ . Durch das konstante  $w_\gamma$  ist die Dichteverteilung über  $w_x$  und  $w_y$  konstant (Abbildung 4.3(a)) und somit leichter zu veranschaulichen. Es ist deutlich zu erkennen, dass in Abbildung 4.3(b) - (d) Partikel nur an Stellen hoher Wahrscheinlichkeit überleben und schließlich gegen die „wahre“ Position konvergieren. Durch den Resampling-Prozess werden immer wieder die Partikel mit hohen Wahrscheinlichkeiten ausgewählt und so stellt sich die Verteilung in Abbildung 4.3(d) ein. Der Partikel-Filter ist im Gegensatz zum Kalman-Filter in der Lage jegliche Art einer Dichteverteilung abzubilden. In Abbildung 4.4 ist beispielsweise eine Dichteverteilung mit mehreren Maxima dargestellt. Dabei werden sich Partikel vorrangig in Gebieten hoher Wahrscheinlichkeit konzentrieren.



**Abbildung 4.3:** Initialisierung des Partikel-Filters mit 1000 Partikeln in einem  $10m \times 10m$  großen Raum mit konstantem  $w_\gamma$ : (a) Dichtefunktion der Partikelverteilung und Initialisierung der Partikel nach (b)  $t = 2$ , (c)  $t = 5$  und (d)  $t = 15$  Zeitschritten

Der Resampling-Prozess muss nicht zwingend ein eindeutiges Ergebnis liefern. So können durch Symmetrien in der Landmarkenanordnung mehrere Positionen mit hoher Wahrscheinlichkeit existieren. Diese werden aber spätestens durch die Bewegung des Fahrzeuges zu der „wahren“ Position konvergieren.

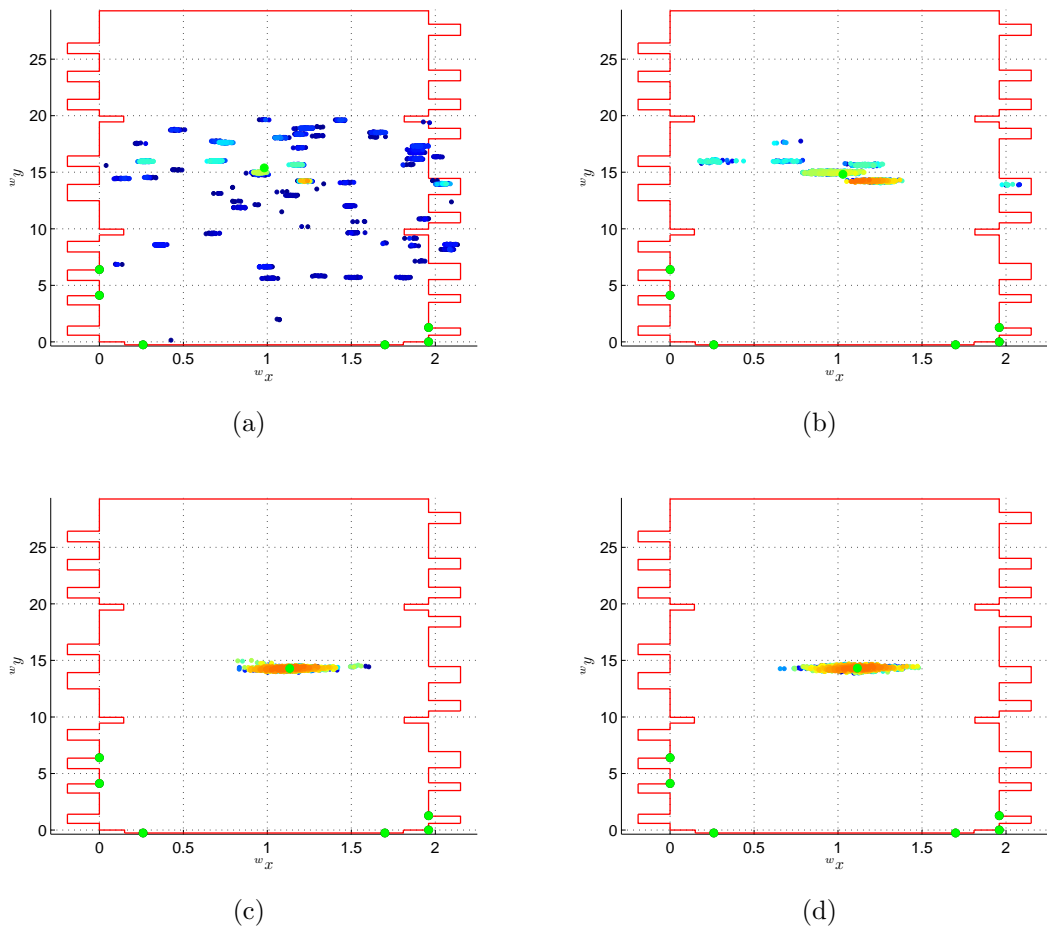


**Abbildung 4.4:** Dichteverteilung mit mehreren Maxima eines Partikel-Filters in einem  $10m \times 10m$  großen Raum mit konstantem  $w_\gamma$

In Abbildung 4.5 wurde eine Initialisierung in einem etwa  $2m \times 30m$  langem Gang mit einem Fahrzeug durchgeführt. Dabei ist auch hier gut zu erkennen, wie sich die Partikel in Bereichen hoher Wahrscheinlichkeit sammeln und schließlich zu einer Position konvergieren. Unter realen Bedingungen hängt natürlich die Initialisierung auch von der Vermessung der Landmarken ab, was durch die Messkovarianzen zum Ausdruck gebracht wird.

#### 4.1.1 Einfluss der Messkovarianzen

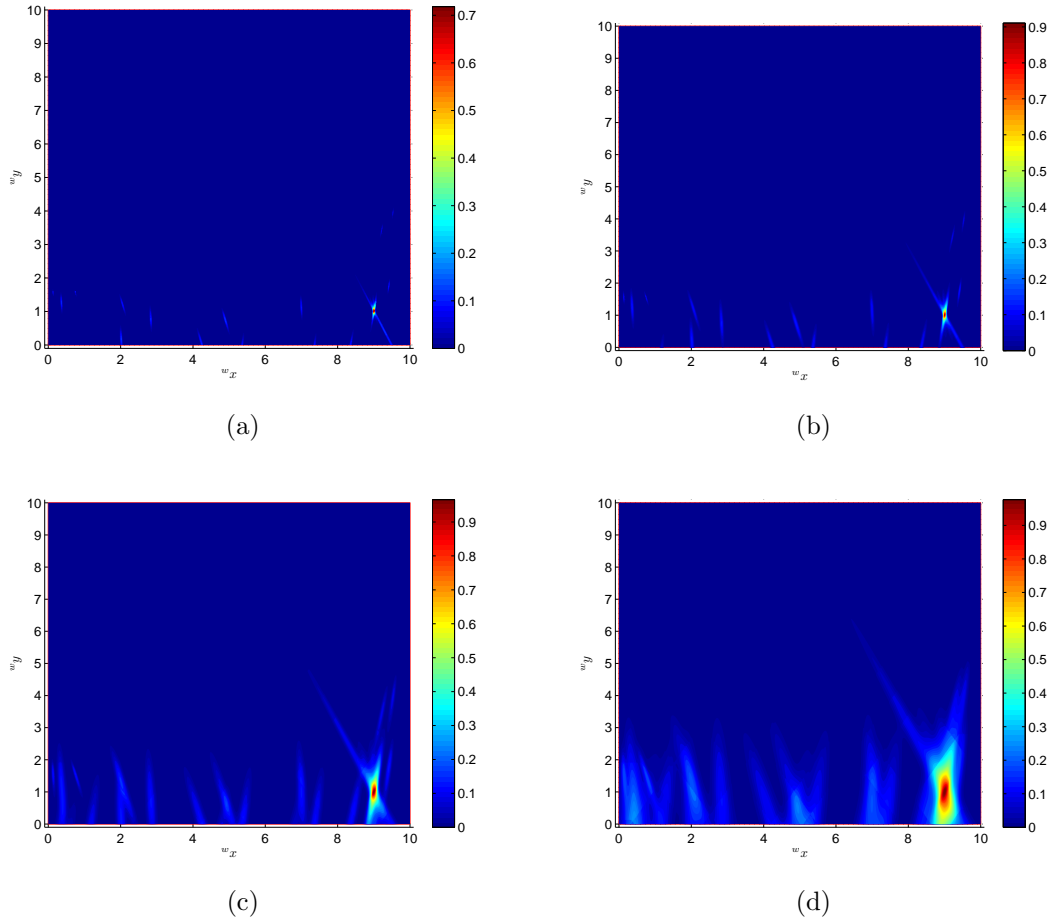
Der Einfluss der Messkovarianzen des Erweiterte-Kalman-Filters auf den Initialisierungsprozess beschreibt das Einschwingverhalten zu dem Initialzustand  $est(x_0)$ . Das bedeutet, wie schnell sich dieser der eigentlichen Schätzung annähert. Unter



**Abbildung 4.5:** Darstellung einer realen Initialisierung in einem etwa  $2m \times 30m$  langen Gang mittels Partikel-Filter mit 2000 Partikeln nach (a)  $t = 2$ , (b)  $t = 10$ , (c)  $t = 50$  und (d)  $t = 80$  Zeitschritten



Berücksichtigung des Sensormodells und der darin enthaltenen Abbildungen wurde ein Sigma  $\sigma_x = \sigma_y = 5px$  gewählt.



**Abbildung 4.6:** Einfluss der Messkovarianzen bei Initialisierung des Partikel-Filters in einem  $10m \times 10m$  großen Raum mit konstantem  $w_\gamma$  mit (a)  $\sigma_R = 1px$ , (b)  $\sigma_R = 2px$ , (c)  $\sigma_R = 5px$  und (d)  $\sigma_R = 10px$

Im Falle des Partikel-Filters haben die Messkovarianzen Einfluss auf die Wahrscheinlichkeitsdichteverteilung und somit auf den Resampling-Prozess der Partikel. Abbildung 4.6 stellt eine Dichteverteilung bei konstanter Ausrichtung ( $w_\gamma = 90^\circ$ ) für verschiedene Standardabweichungen  $\sigma_R$  dar. Dabei wird deutlich, dass bei Vergrößerung der Messkovarianz die Gebiete hoher Wahrscheinlichkeit zunehmen. Somit ist es für die Initialisierung wahrscheinlicher, dass sich Partikel in diesen Bereichen anhäufen. Jedoch vergrößert sich auch die Partikelwolke bei steigender Standardabweichung und somit auch die Unsicherheit der Schätzung  $est(x_t)$ . Für den in der

Aufgabe verwendeten Partikel-Filter unter Berücksichtigung des Sensormodells wurde ein Sigma  $\sigma_R = 5px$  verwendet.

## 4.2 Bewegungsmodell

In diesem Abschnitt werden die Fahrzeugbewegungsmodelle (Systemmodelle) beider Filter miteinander verglichen. Dabei wird das Prädiktionsverhalten für verschiedene Strecken (Diagonal- und Kurvenfahrt) untersucht. Es wird für beide Filter ein Initialzustand eingestellt. Im Falle des Kalman-Filters wird der Korrekturschritt unterbunden (Nichtvorhandensein von Messungen) und der Bahnverlauf wird nur durch das Modell bestimmt. Der Filter wird durch den Modalwert der Vorhersage  $\widetilde{est}(x_t)$  und der Unsicherheit, die durch das Gebiet (Ellipse) mit der höchsten Wahrscheinlichkeit dargestellt ist, beschrieben. Dieses Gebiet entspricht dabei einer Höhenlinie der Normalverteilung  $N(\hat{x}_t, P_t)$  bestimmt durch die Kovarianzmatrix  $P_t$ .

Der Partikel-Filter wird durch den Modalwert seiner Verteilung  $\widetilde{est}(x_t)$  und den gleichverteilten Partikeln beschrieben. Durch den verwendeten Resampling-Prozess wird nichts an dieser Verteilung geändert, da auch hier keine Messungen betrachtet werden, sondern nur das Systemverhalten untersucht wird.

Die Startungenauigkeiten beider Filter werden dabei mit Null angenommen, so dass nur die Verunschärfung in das System einfließt. Als Zeitdifferenz zwischen zwei Prädiktionen wird aufgrund der besseren Darstellung eine Sekunde gewählt.

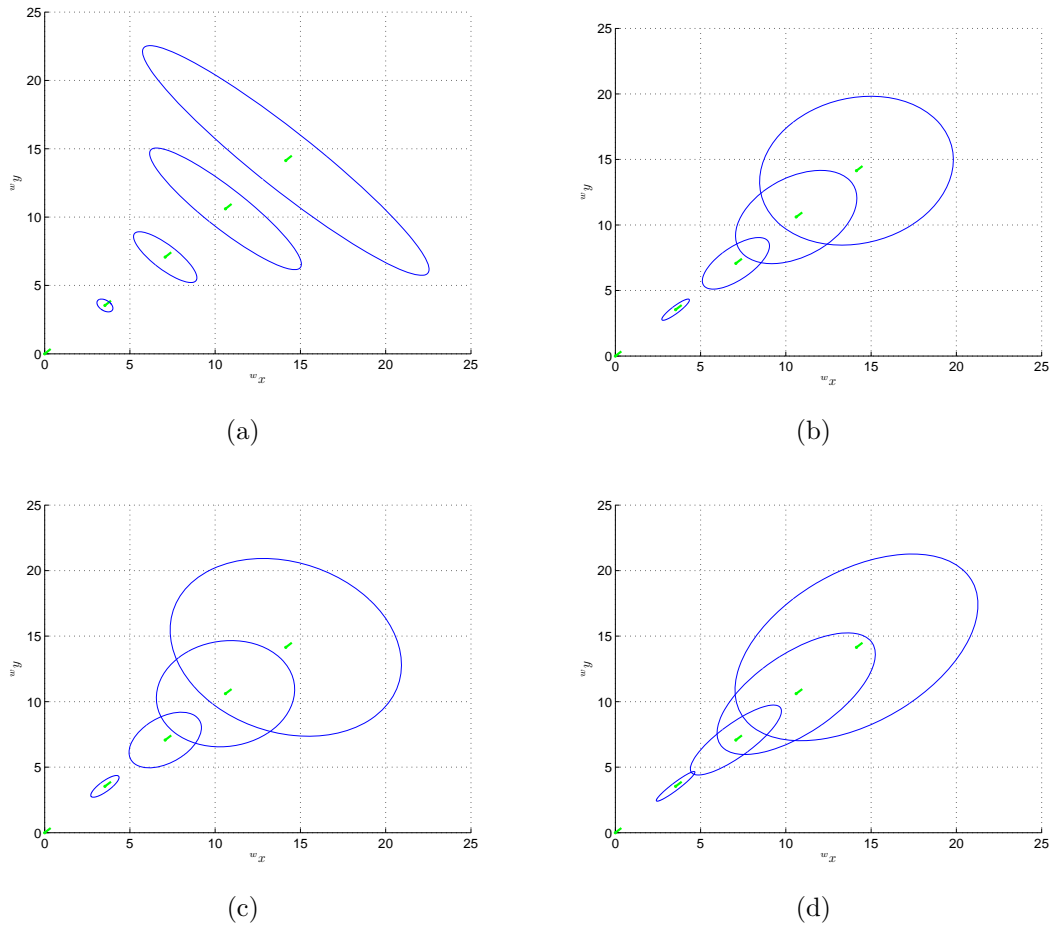
### 4.2.1 Diagonalfahrt

Abbildung 4.7 zeigt eine Diagonalfahrt für den Kalman-Filter. Dabei wird der Initialzustand so eingestellt, dass sich eine solche Fahrt ergibt.

$$\hat{x}_0 = \left[ x_0 = 0 \text{ m}, y_0 = 0 \text{ m}, \gamma = \frac{\pi}{4} \text{ rad}, v_0 = 5 \frac{\text{m}}{\text{s}}, \omega_0 = 0 \frac{\text{rad}}{\text{s}} \right]^T$$

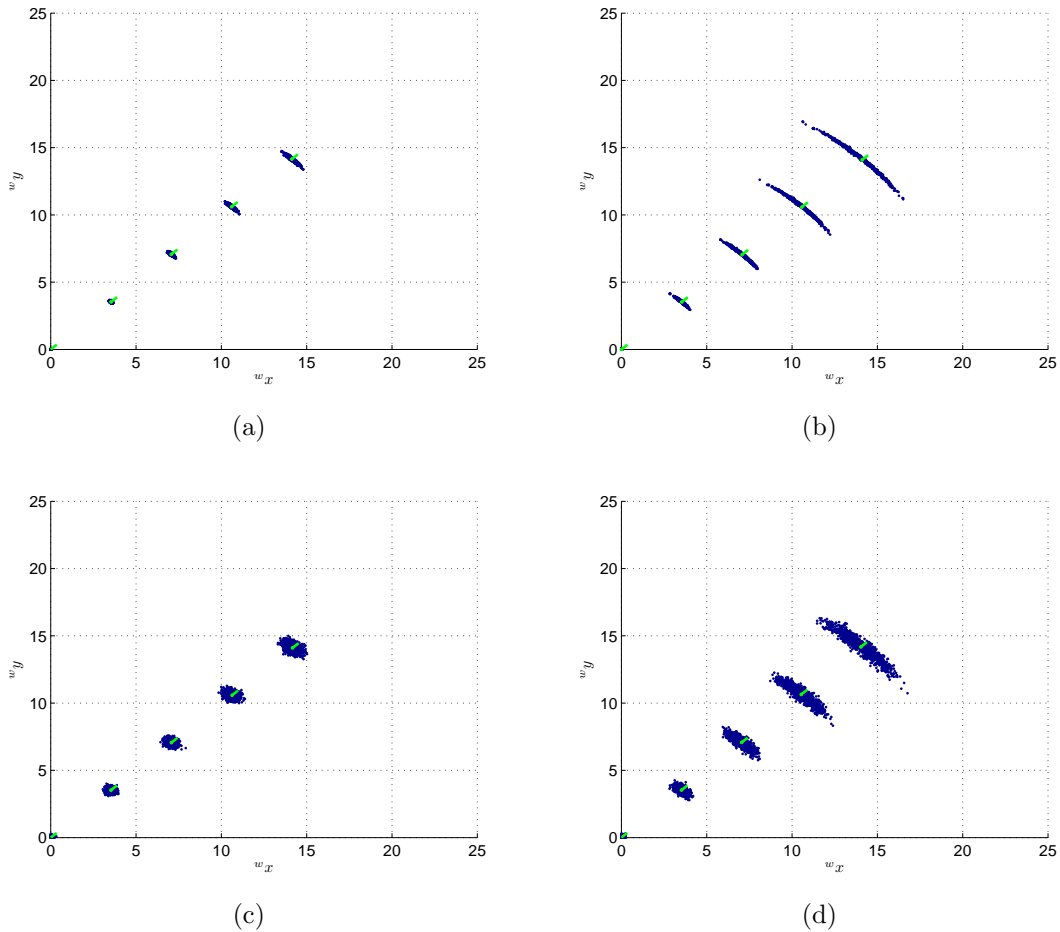
Das CTRV-Modell stellt dabei eine auf die Bewegungsrichtung abgestimmte Unsicherheit aufgrund der Unterscheidung in Längs- und Querkomponente dar. Es ist deutlich zu erkennen, wie sich die Standardabweichung der Längs- und Querkomponente auf die Form der Ellipse auswirkt.

Die in Abbildung 4.8 dargestellte Diagonalfahrt eines Partikel-Filters spiegelt dessen



**Abbildung 4.7:** Darstellung der Position und Unsicherheit des Bewegungsmodells für eine Diagonalfahrt mit (a)  $\sigma_v = 0,1 \text{ m/s}$ ,  $\sigma_\omega = 0,05 \text{ rad/s}$ , (b)  $\sigma_v = 1,0 \text{ m/s}$ ,  $\sigma_\omega = 0,01 \text{ rad/s}$ , (c)  $\sigma_v = 1,0 \text{ m/s}$ ,  $\sigma_\omega = 0,02 \text{ rad/s}$  und (d)  $\sigma_v = 2,0 \text{ m/s}$ ,  $\sigma_\omega = 0,01 \text{ rad/s}$

Prädiktionsverhalten bezüglich unterschiedlicher Standardabweichungen wieder. Dabei haben die Unsicherheiten  $\sigma_x$  und  $\sigma_y$  Einfluss auf die Längskomponente der Verteilung und  $\sigma_\gamma$  auf die Querkomponente. Jedoch sind die Unsicherheiten im Vergleich zum Kalman-Filter nicht so stark ausgeprägt.



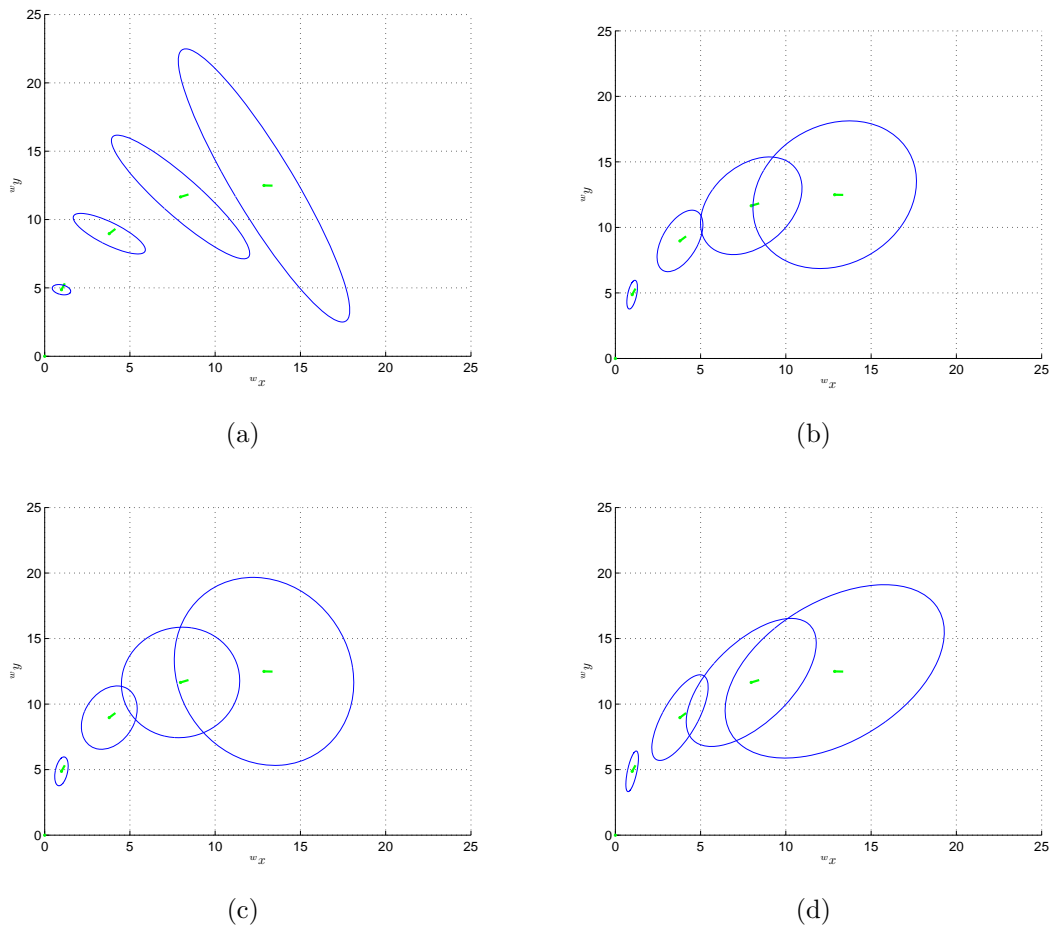
**Abbildung 4.8:** Darstellung der Position und Unsicherheit (repräsentiert durch Partikel) des Bewegungsmodells für eine Diagonalfahrt mit (a)  $\sigma_x = 0,02 m$ ,  $\sigma_y = 0,02 m$ ,  $\sigma_\gamma = 0,01 rad$ , (b)  $\sigma_x = 0,02 m$ ,  $\sigma_y = 0,02 m$ ,  $\sigma_\gamma = 0,05 rad$ , (c)  $\sigma_x = 0,1 m$ ,  $\sigma_y = 0,1 m$ ,  $\sigma_\gamma = 0,01 rad$  und (d)  $\sigma_x = 0,1 m$ ,  $\sigma_y = 0,1 m$ ,  $\sigma_\gamma = 0,05 rad$

Es ist deutlich zu erkennen, wie sich eine Vergrößerung (Abbildung 4.8(b) und 4.8(d)) von  $\sigma_x, \sigma_y$  und  $\sigma_\gamma$  auf die Form der Partikelwolke auswirkt. Wobei die Querkomponente einen deutlich größeren Einfluss hat als die Längskomponente.

### 4.2.2 Kurvenfahrt

In Abbildung 4.9 ist eine Kurvenfahrt für den Kalman-Filter dargestellt. Dabei wird als Initialzustand folgender verwendet:

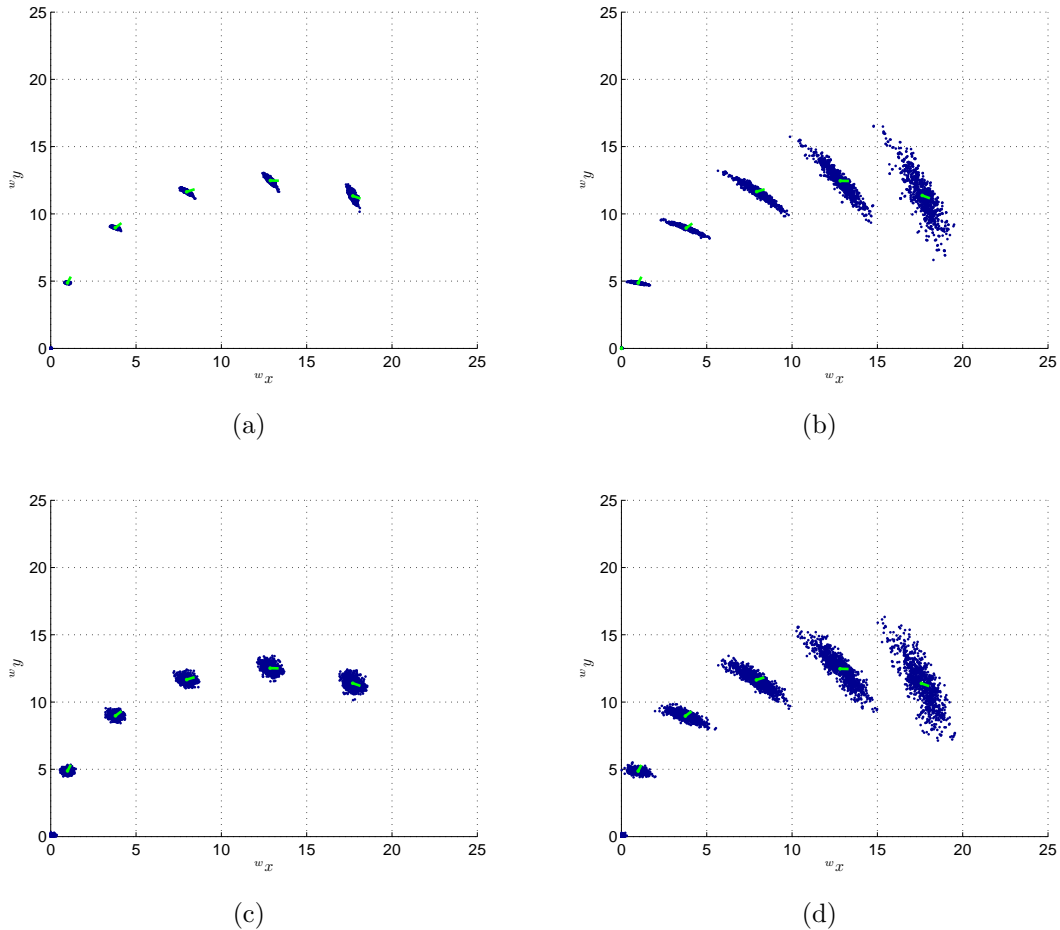
$$\hat{x}_0 = \left[ x_0 = 0 \text{ m}, y_0 = 0 \text{ m}, \gamma = \frac{\pi}{2} \text{ rad}, v_0 = 5 \frac{\text{m}}{\text{s}}, \omega_0 = -0.4 \frac{\text{rad}}{\text{s}} \right]^T$$



**Abbildung 4.9:** Darstellung der Position und Unsicherheit des Bewegungsmodells für eine Kurvenfahrt mit (a)  $\sigma_v = 0,1 \text{ m/s}$ ,  $\sigma_\omega = 0,05 \text{ rad/s}$ , (b)  $\sigma_v = 1,0 \text{ m/s}$ ,  $\sigma_\omega = 0,01 \text{ rad/s}$ , (c)  $\sigma_v = 1,0 \text{ m/s}$ ,  $\sigma_\omega = 0,02 \text{ rad/s}$  und (d)  $\sigma_v = 2,0 \text{ m/s}$ ,  $\sigma_\omega = 0,01 \text{ rad/s}$

Die für das CTRV-Modell verwendeten Unsicherheiten haben dabei verschiedene Auswirkungen. So dominiert, wie auch schon bei der Diagonalfahrt, bei zunehmender

Kurvenlänge eher der Fehler in Querrichtung (Abbildung 4.9(a) und 4.9(d)) und bestimmt somit auch das Aussehen der Ellipse. Die für die Aufgabe verwendeten Unsicherheiten wurden mit  $\sigma_v = 0,1 \text{ m/s}$  und  $\sigma_\omega = 0,05 \text{ 1/s}$  gewählt.



**Abbildung 4.10:** Darstellung der Position und Unsicherheit (repräsentiert durch Partikel) des Bewegungsmodells einer Kurvenfahrt mit (a)  $\sigma_x = 0,02 \text{ m}$ ,  $\sigma_y = 0,02 \text{ m}$ ,  $\sigma_\gamma = 0,01 \text{ rad}$ , (b)  $\sigma_x = 0,02 \text{ m}$ ,  $\sigma_y = 0,02 \text{ m}$ ,  $\sigma_\gamma = 0,05 \text{ rad}$ , (c)  $\sigma_x = 0,1 \text{ m}$ ,  $\sigma_y = 0,1 \text{ m}$ ,  $\sigma_\gamma = 0,01 \text{ rad}$  und (d)  $\sigma_x = 0,1 \text{ m}$ ,  $\sigma_y = 0,1 \text{ m}$ ,  $\sigma_\gamma = 0,05 \text{ rad}$

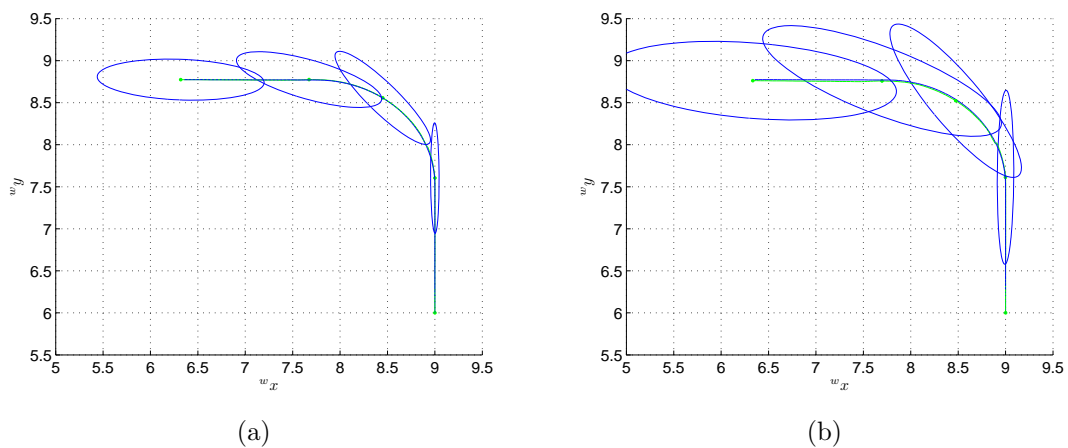
Bei einer Kurvenfahrt des Partikel-Filters zeigt sich nochmals deutlich das Verhalten von  $\sigma_\gamma$  auf die Form der Partikelwolke. Dabei ist im Vergleich zum Kalman-Filter zu bemerken, dass der Einfluss von  $\sigma_x$  und  $\sigma_y$  auf die Längskomponente der Partikelwolke eher gering ist. Jedoch zeigt sich auch hier, dass die Querkomponente bei zunehmender Kurvenlänge dominierend ist (Abbildung 4.10(b) und 4.10(d)). Für den Partikel-Filter

wurden die Standardabweichungen  $\sigma_x = \sigma_y = 0,02 \text{ m}$ ,  $\sigma_\gamma = 0,01 \text{ rad}$  verwendet.

### 4.2.3 Einfluss der Messkovarianzen

Um den Einfluss der Messkovarianzen auf die Filter zu beschreiben, wurde eine Kurvenfahrt mit Hilfe eines Referenzfilters<sup>2</sup> simuliert und die entstehenden Daten (Messungen und Stellgrößen) als Eingabe für den EKF und den Partikel-Filter verwendet.

Für den Kalman-Filter wurden zwei verschiedene Unsicherheiten verwendet. In Abbildung 4.11 ist das Ergebnis der Kurvenfahrt dargestellt. Dabei wurde für bestimmte Zeitpunkte die Unsicherheit der Schätzung, repräsentiert durch eine Ellipse (für die Darstellung wurde die Ellipse stark vergrößert), abgebildet. Es ist zu erkennen, dass eine Vergrößerung der Messungenauigkeit von  $\sigma_\omega$  auch zu einer Vergrößerung der Ungenauigkeit bei der Kurvenfahrt führt.

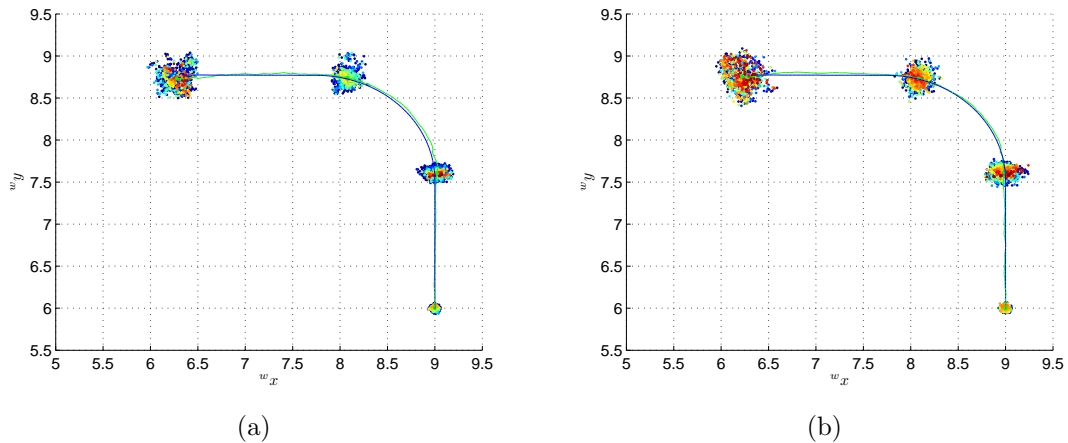


**Abbildung 4.11:** Darstellung des Einflusses zwei verschiedener Unsicherheiten auf das Fahrverhalten des Kalman-Filters mit (a)  $\sigma_x = 5 \text{ px}$ ,  $\sigma_y = 5 \text{ px}$  und (b)  $\sigma_x = 10 \text{ px}$ ,  $\sigma_y = 10 \text{ px}$

Im Falle des Partikel-Filters ist in Abbildung 4.12 nochmals der Einfluss der Messkovarianzen auf die Partikelwolke dargestellt. Dabei sieht man deutlich, wie eine Vergrößerung von  $\sigma_R$  sich auf die Verteilung der Partikelwolke auswirkt. Bei größerem  $\sigma_R$  (Abbildung 4.12(b)) sind deutlich mehr Partikel mit hoher Wahrscheinlichkeit vorhanden als bei niedrigem  $\sigma_R$ . Im Vergleich zum Kalman-Filter wirkt sich eine

<sup>2</sup>engl. *ground-truth filter*

Vergrößerung der Messkovarianzen nur auf die Verteilung und somit auf die Dichtefunktion für das Resampling aus. Beim Kalman-Filter hingegen vergrößert sich die Schätzung (größere Ellipsen), was durchaus zu einem Fehlverhalten des Filters führen kann.



**Abbildung 4.12:** Darstellung des Einflusses zwei verschiedener Messkovarianzen auf das Fahrverhalten des Partikel-Filters mit (a)  $\sigma_R = 5 \text{ px}$  und (b)  $\sigma_R = 10 \text{ px}$

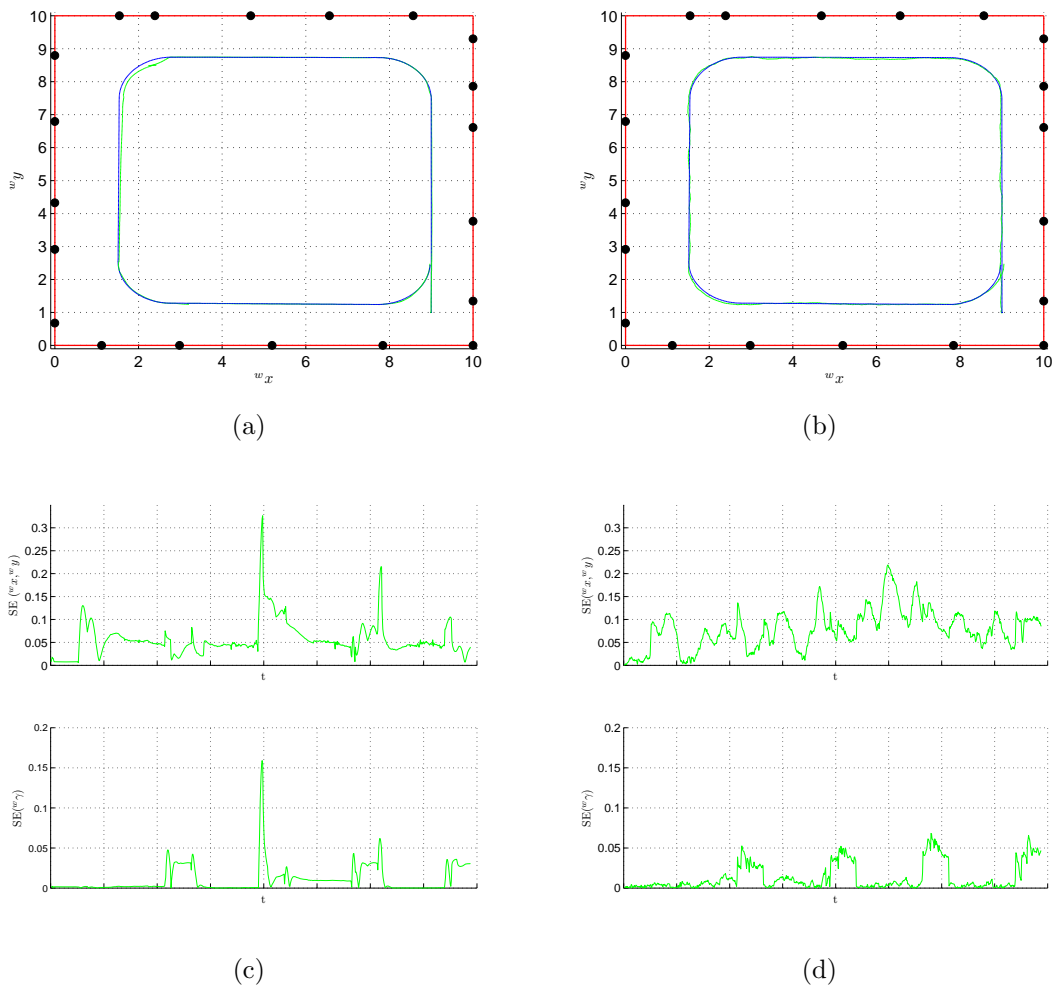
### 4.3 TestszENARIO

In diesem Abschnitt werden simulierte und praktische Ergebnisse ausgewertet und somit beide Filter verglichen. Die praktische Testfahrt wurde dabei in einem etwa  $2m \times 30m$  langem Gang, der mit Landmarken bestückt war, durchgeführt.

Abbildung 4.13 stellt eine simulierte Fahrt in einem  $10m \times 10m$  großen Raum dar. In diesem Raum befinden sich an jeder Wand Landmarken (schwarze Punkte), die die Positionsbestimmung ermöglichen. Danach wurden beide Filter initialisiert. Für den Kalman-Filter wurde der Initialzustand  $est(x_0)$  festgelegt. Im Falle des Partikel-Filters wurde eine Initialisierung (Abschnitt 4.1) durchgeführt. Die Messungen wurden dabei mit Hilfe eines Referenzfilters (*ground-truth filter*), dargestellt durch eine blaue Linie, ermittelt. Die Fehlerbetrachtungen beider Filter beziehen sich dabei immer auf den Referenzfilter.

Für beide Filter ergibt sich dabei annähernd der gleiche Verlauf (Abbildung 4.13(a) und 4.13(b)). Anhand des Fehlerverlaufs ist jedoch ersichtlich, dass sich auch hier





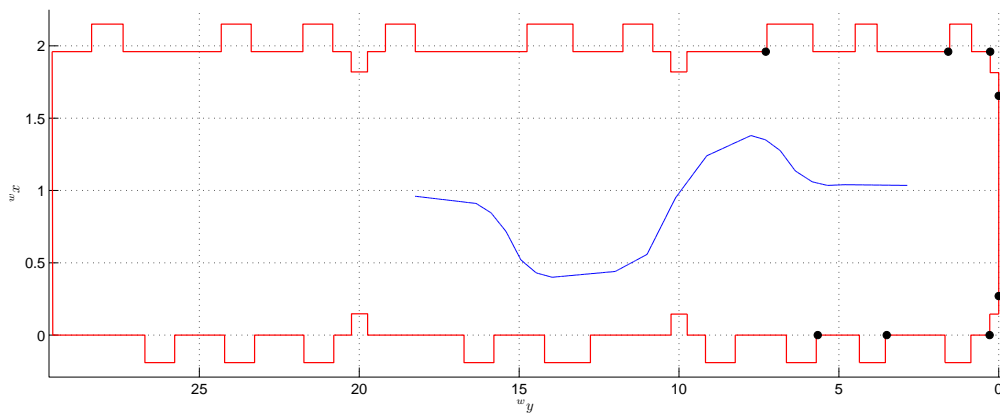
**Abbildung 4.13:** Darstellung der simulierten Testfahrt für (a) Kalman-Filter und (b) Partikel-Filter, sowie die entsprechenden Fehler (Mittlere quadratische Abweichung) bezüglich des Referenzfilters (blaue Linie) für (c) Kalman-Filter und (d) Partikel-Filter

geringfügige Abweichungen bedingt durch Systemfehler und Messkovarianzen ergeben.

	EKF	PARTIKEL-FILTER
Mittlere Fehler von ${}^w x, {}^w y$	0,0675 m	0,0902 m
Mittlere Fehler von ${}^w \gamma$	0,0205 rad	0,0215 rad

**Tabelle 4.1:** Darstellung der mittleren quadratischen Abweichung einer simulierten Testfahrt für die Zustandsvariablen  ${}^w x, {}^w y$  und  ${}^w \gamma$  beider Filter

In Tabelle 4.1 ist der mittlere quadratische Fehler<sup>3</sup> für die Zustandsvariablen  ${}^w x, {}^w y$  und  ${}^w \gamma$  beider Filter angegeben. Dabei unterscheiden sich diese nur geringfügig. Die Fehler entstehen sowohl durch die System- und Sensormodellfehler als auch durch das sprunghafte Verhalten des Referenzfilters in den Kurven, welches auch deutlich in Abbildung 4.13(c) und 4.13(d) zu erkennen ist.



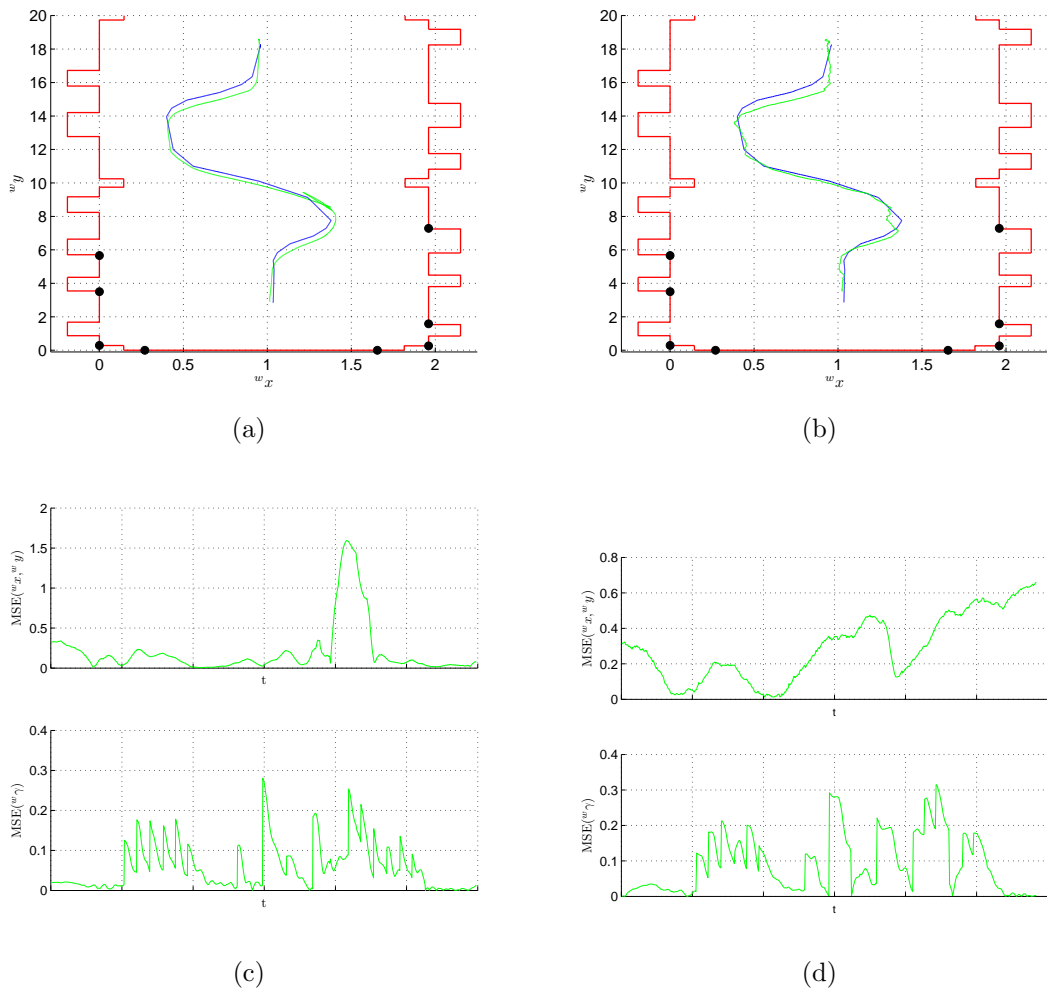
**Abbildung 4.14:** Darstellung eines  $2m \times 30m$  langen Ganges mit Referenzspur (blaue Linie)

Die praktische Testfahrt wurde mit dem in Kapitel 1 vorgestellten Fahrzeug durchgeführt. Dieses Fahrzeug ist mit einem Line-Sensor ausgestattet, so dass es einer vorgegebenen Spur folgen kann. Die Aufnahme der Daten erfolgte dabei in einem  $2m \times 30m$  langem Gang (Abbildung 4.14), in dem eine ausgemessene Referenzspur aufgeklebt war.

Mit Hilfe des FTS (inklusive Inkrementalgeber und Drehratensensor) und der Referenzlinie (*Groundtruth*) wurden die in Abbildung 4.15(a) und 4.15(b) dargestellten Ergebnisse erzielt. Auch hier ergibt sich für beide Filter annähernd der gleiche Verlauf.

<sup>3</sup>engl. *mean square error (MSE)*

In Tabelle 4.2 sind die quadratischen Abweichungen der Zustandsvariablen dargestellt. Allerdings ist hier der Fehler der Zustandsvariable  $w_y$  deutlich größer als bei der Simulation. Dies liegt vor allem in der relativ ungenauen Groundtruth des Fahrzeuges, da die Referenzposition nur mit Hilfe der Odometriedaten ermittelt wurde. Die Abweichung der Zustandsvariablen  $w_x, w_\gamma$  ist bei beiden Filtern ungefähr gleich und auch nicht zu groß. Somit unterscheiden sich die beiden Filter bezüglich der Genauigkeit kaum voneinander. Die Robustheit des in dieser Aufgabe verwendeten Kalman-Filters



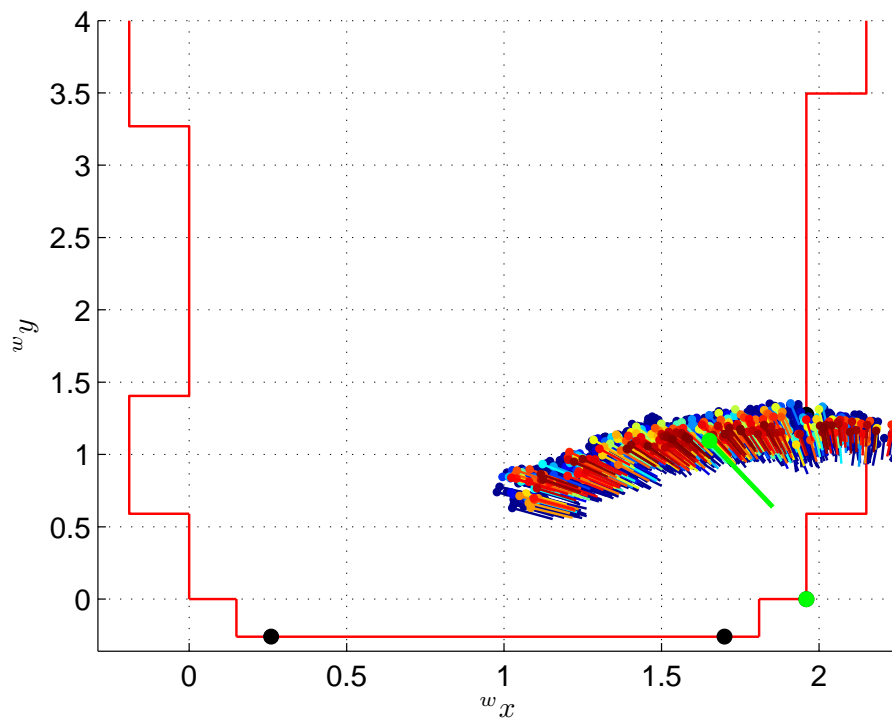
**Abbildung 4.15:** Darstellung einer realen Testfahrt für (a) Kalman-Filter und (b) Partikel-Filter, sowie die entsprechenden Fehler (Mittlere quadratische Abweichung) bezüglich des Referenzfilters (blaue Linie) für (c) Kalman-Filter und (d) Partikel-Filter

ist aus praktischer Sicht nicht ausreichend. Dies ergibt sich aus den Zuordnungsme-

	EKF	PARTIKEL-FILTER
Mittlere Fehler von $w_x$	0,0621 m	0,0848 m
Mittlere Fehler von $w_y$	0,3816 m	0,3337 m
Mittlere Fehler von $w_\gamma$	0,0811 rad	0,1233 rad

**Tabelle 4.2:** Darstellung der mittleren quadratischen Abweichung einer praktischen Testfahrt für die Zustandsvariablen  $w_x, w_y$  und  $w_\gamma$  beider Filter

thoden des Sensormodells, da hier eine falsche Zuordnung einen großen Einfluss auf die Schätzung hat und somit auch schnell zu falschen Ergebnissen führen kann. Im Vergleich dazu ist der Partikel-Filter anpassungsfähiger. Aufgrund der Partikelwolke kann sich dieser besser an Zustandsänderungen anpassen und auch Fehlmessungen (Fehlererkennung von Landmarken) haben im Vergleich zum Kalman-Filter keinen zu großen Einfluss auf das Ergebnis der Schätzung. Allerdings ist der Rechenaufwand für den Partikel-Filter viel größer als der des Kalman-Filters.



**Abbildung 4.16:** Darstellung der Partikel-Filter-Verteilung, wenn sich nur eine Landmarke im Sichtbereich des Filters befindet

Abbildung 4.16 stellt abschließend die Verteilung eines Partikel-Filters dar, wobei

sich nur eine Landmarke im Sichtbereich des Filters befindet. Diese Zuordnung ist nicht eindeutig auf eine Position zurückzuführen. Dieser Fakt wird auch in der Verteilung deutlich zum Ausdruck gebracht. Es bildet sich eine Art Halbkreis um diese Landmarke. Eine solche Verteilung kann im Vergleich zum Kalman-Filter nur der Partikel-Filter beschreiben.

## 5 Software

Als softwaretechnische Realsierung wurde ein Programm („Landmark Locator“) entwickelt. Mit Hilfe dieses Programmes lassen sich folgende Funktionen durchführen:

- Erkennung und Anzeige von Landmarken
- Online/Offline Funktionalität unter Verwendung des Kalman- oder Partikel-Filters
- Simulation von Szenarien
- Laden vorgefertigter Karten
- Anzeige dieser Karten und der Filterergebnisse

Die Applikation und die verwendeten Bibliotheken wurde dabei modular aufgebaut und mit dem Ziel einer schnellen Verarbeitung entwickelt. Die entstandenen Bibliotheken können somit auch in Embedded-Rechnersystemen mit entsprechender Rechenleistung eingesetzt werden. Als Programmiersprache kam dabei für die grafische Benutzerschnittstelle<sup>1</sup> C++/cli (*.Net-Framework*) und für die verwendeten Bibliotheken C++ zum Einsatz. Desweiteren wurden auch einige an der Professur für Nachrichtentechnik bereitgestellte Bibliotheken verwendet. Im folgenden Abschnitt wird die Struktur des Gesamtsystems sowie einige Einzelkomponenten kurz erläutert.

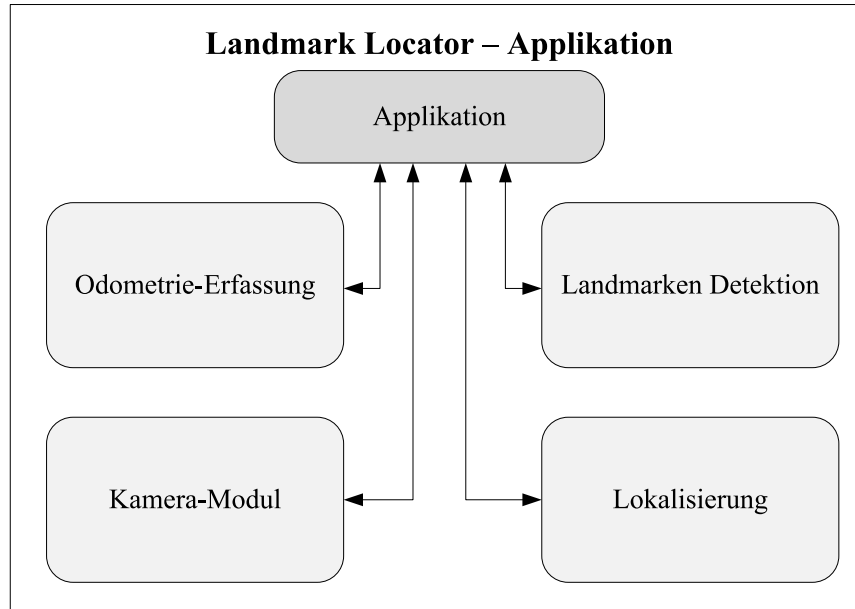
### 5.1 Struktur der Applikation

Die Struktur der Applikation gliedert sich in verschiedene Komponenten, die jeweils eine bestimmte Aufgabe übernehmen. Diese Komponenten sind in Abbildung 5.1 schematisch dargestellt. Dabei laufen die Informationen aus den jeweiligen Komponenten in der eigentlichen Applikation zusammen und werden weiterverarbeitet. Die

---

<sup>1</sup>engl. *Graphical User Interface (GUI)*

einzelnen Komponenten wurden als Programmbibliothek<sup>2</sup> implementiert, um eine Wiederverwendbarkeit der Module zu erreichen.



**Abbildung 5.1:** Schematische Darstellung der einzelnen Komponenten der Applikation „Landmark Locator“

## Odometrie-Erfassung

Die Odometrie-Komponente hat die Aufgabe die Daten des Inkrementalgebers (Abschnitt 2.1.2) und des Drehratensensors (Abschnitt 2.1.3) bereitzustellen. Diese Informationen werden dabei als Stellgrößen von dem Lokalisierungsmodul für den Partikel-Filter verwendet.

## Kamera-Modul

Die Kamerakomponente erfasst die Infrarotbilder der in Abschnitt 2.1.1 beschriebenen Kamera (AVT Guppy) und stellt diese dann zur weiteren Verarbeitung zur Verfügung. Mit Hilfe dieser Infrarotbilder sind die verwendeten Landmarken (Abschnitt 2.2) besonders gut sichtbar und können so von der Landmarken-Detektion extrahiert werden. Desweiteren ist es möglich, Bilder für eine spätere Offline-Verarbeitung

<sup>2</sup>engl. *Dynamic Link Library* (DLL)

abzuspeichern. Dies ist oft notwendig, um Filter mit realen Daten testen zu können und ihre Ergebnisse näher zu betrachten.

## Landmarken-Detektion

Die Landmarken-Detektions-Komponente ist von großer Bedeutung, da diese die Landmarken aus den Infrarotbildern extrahiert und der Lokalisierung zur Verfügung stellt. Diese Komponente wurde für eine schnelle Extraktion der Landmarken optimiert (Rechenzeit ca. 1 *ms*). Dabei werden in dem Algorithmus nur die hellen Bereiche des Bildes betrachtet und entsprechend zu einer Detektion zusammengefasst.

## Lokalisierungsmodul

Das Lokalisierungsmodul beinhaltet die Implementierung der in Kapitel 3 beschriebenen Filter. Als Erweiterte-Kalman-Filter wurde der von der Professur für Nachrichtentechnik zur Verfügung gestellte Filter verwendet. Der Partikel-Filter wurde als Bibliothek, wie in Abschnitt 3.3 beschrieben, erstellt und in das Modul integriert. Außerdem beinhaltet dieses Modul einen Simulationsfilter (Groundtruth-Filter), mit dem es möglich ist eine simulierte Testfahrt zu erstellen, um die Filter unter idealen Bedingungen zu testen.

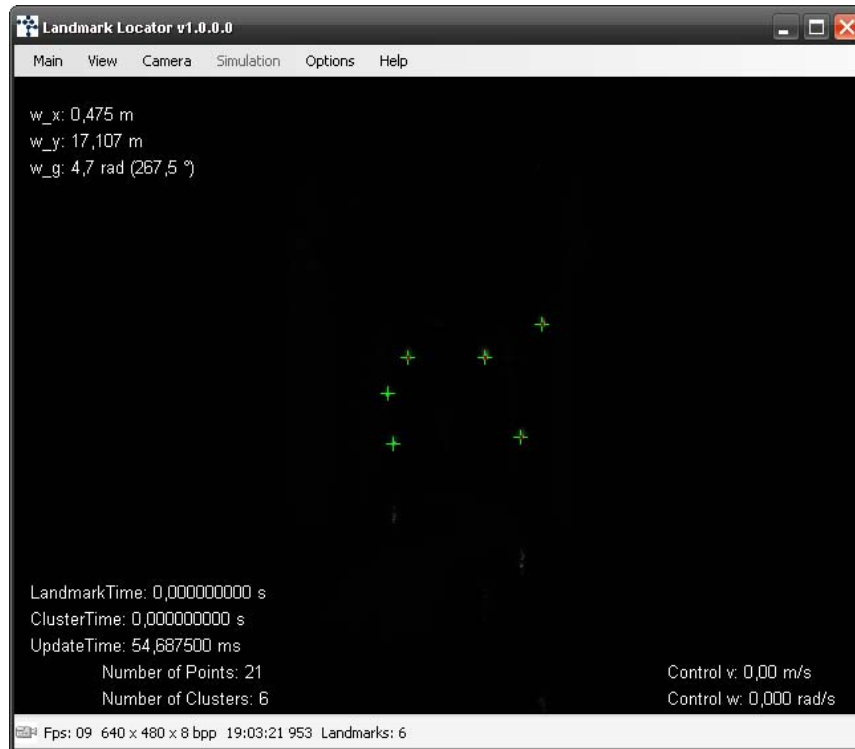
## Applikation „Landmark Locator“

Die Applikation verbindet alle beschriebenen Komponenten miteinander und stellt die Ergebnisse grafisch dar. Über ein *User Interface* (GUI) können die verschiedenen Funktionen aufgerufen werden. Die entsprechenden Funktionen sind:

- On- oder Offline-Verarbeitung unter Verwendung eines Filters oder Durchführung einer Simulation
- Laden von Karten- und Szenarioinformationen, um die simulierte/reale Umgebung besser beschreiben und Simulationen durchführen zu können
- Exportieren verschiedener Daten (Schätzung, Partikel, etc.) um diese für andere Anwendungen zur Verfügung zu stellen
- verschiedene Einstellungsmöglichkeiten bezüglich der verwendeten Filter



In Abbildung 5.2 ist die Anwendung „Landmark Locator“ dargestellt. Dabei wird gerade ein Bild der Kamera verarbeitet und verschiedene Informationen (z.B. Landmarken (grüne Kreuze)) angezeigt.



**Abbildung 5.2:** Screenshot des GUI der Anwendung „Landmark Locator“

Desweiteren wurde für die räumliche Anzeige einer Karte und der darin enthaltenen Landmarken eine 2D-Anzeige erstellt. In dieser Anzeige wurde auch die Schätzung der Filter bzw. die Darstellung der Partikel realisiert. Für die 2D-Anzeige wurde als Programmierschnittstelle DirectX<sup>3</sup> verwendet. Da diese Technik sehr auf die Grafik-Hardware abgestimmt ist, stehen für die übrige Anwendung noch genügend Ressourcen zur Verfügung. Allerdings setzt diese Technik doch etwas leistungsfähigere Rechner voraus. Aus diesem Grund wird die 2D-Anzeige nur für die grafische Darstellung der Ergebnisse verwendet.

In Abbildung 5.3 ist die 2D-Anzeige des „Landmark Locator“ beispielhaft für eine Simulation des Partikel-Filters (Initialisierungsvorgang) dargestellt. Darin sind gut

<sup>3</sup>DirectX ist eine Sammlung von Software-Komponenten, die in dem Multimedienbereich (vorrangig 2D- und 3D-Grafik) Anwendung findet.

die Kartengrenzen (grüne Line) und die Landmarken (weiße Punkte) zu erkennen. Desweiteren ist die Schätzung des Partikel-Filters (grüner Punkt) sowie die Partikel selbst und der Groundtruth-Filter (blauer Punkt) abgebildet.

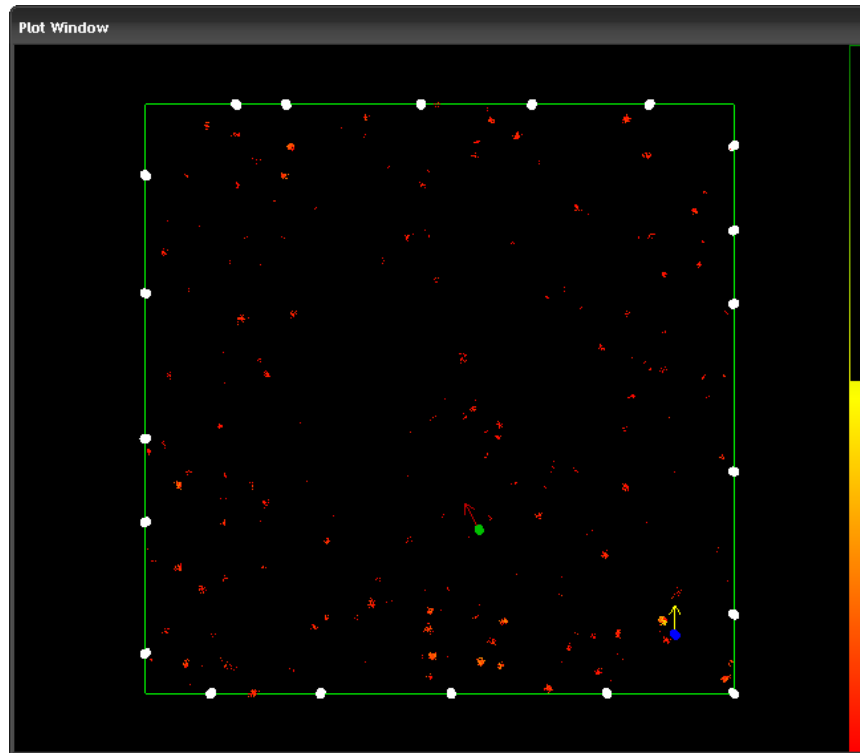


Abbildung 5.3: Screenshot der 2D-Anzeige der Anwendung „Landmark Locator“

## 6 Zusammenfassung

In diesem abschließenden Kapitel wird die Arbeit zusammengefasst, sowie ein Ausblick auf weiterführende Arbeiten gegeben.

### 6.1 Ergebnisse

Das Ergebnis dieser Arbeit stellt die entstandene Software dar. Diese ist in der Lage ein Fahrzeug mittels des in dieser Arbeit vorgestellten Partikel-Filters und in der Umgebung befindlichen Landmarken zu lokalisieren. Die Landmarken wurden dabei aus einem Infrarotbild einer Kamera extrahiert und als Messung durch den Filter verarbeitet. Der Partikel-Filter beschränkt sich nicht auf eine bestimmte A-posteriori-Verteilung sondern versucht durch die finite Anzahl an Partikeln jede mögliche Verteilung zu approximieren. Desweiteren wurde der Partikel-Filter mit dem Erweiterten-Kalman-Filter hinsichtlich der gestellten Aufgabe verglichen.

Es zeigte sich, dass der Partikel-Filter sehr anpassungsfähig und robust in Bezug auf eine Lokalisierung mittels Landmarken ist. Betrachtet man dazu den Erweiterten-Kalman-Filter so beschränkt sich dieser auf eine Gauß'sche Normalverteilung als A-posteriori-Verteilung und es stellt sich häufig durch die falsche Zuordnung von Messungen im Sensormodell ein falsches Schätzergebnis ein. Außerdem ist es dem Erweiterten-Kalman-Filter nicht möglich eine Initialisierung seiner Ausgangsposition durchzuführen. Im Gegenzug dazu kann sich der Partikel-Filter durch eine Verteilung seiner Partikel im Zustandsraum initialisieren. Allerdings sind mit zunehmender Dimension des Zustandsvektors sehr viele Partikel für eine erfolgreiche Initialisierung notwendig. Aus diesem Grund wird stets versucht die Dimension des Zustandsvektors eines Partikel-Filters gering zu halten. Positiv zeigt sich somit auch, dass der Partikel-Filter in der Lage ist eine neue Initialisierung durchzuführen, wenn dieser seine Position verloren haben sollte.

Die Bewertung der vorgestellten Ergebnisse zeigt, dass es mit dem Partikel-Filter sehr gut möglich ist, eine erfolgreiche Initialisierung und eine Schätzung der Position eines Fahrzeuges mittels Landmarken durchzuführen.

## 6.2 Ausblick

Für die vorgestellte Art der Lokalisierung soll an dieser Stelle trotz der Vorteile Verbesserungsideen diskutiert werden. Im Folgenden werden weitere Ansätze für die Weiterführung dieser Arbeit aufgezeigt:

- **KARTEN**  
Durch eine bessere Verknüpfung von Karteninformationen (Ausdehnung der Karte, Begrenzungen, etc.) durch den Filter könnten Zustände, die aufgrund der Lage nicht möglich sind, ausgeschlossen werden.
- **FAHRZEUG**  
Die Ausdehnung des Fahrzeuges sowie mögliche Bewegungen des ganzen Fahrzeuges sollte mit Karteninformationen verknüpft werden, um somit auch mögliche Zustände auszuschließen, die durch die Ausdehnung des Fahrzeuges nicht zutreffend sind.
- **LANDMARKEN**  
Die in Abschnitt 2.2 vorgestellte Beschreibung von Landmarken sollte für eine bessere Lokalisierung berücksichtigt werden.
- **SLAM**  
Eine mögliche Erweiterung eines Lokalisierungsfilters unter Einbeziehung von Karteninformationen wäre das sogenannte SLAM<sup>1</sup>. Dieses Problem beschreibt die Lokalisierung eines Roboters in einer unbekanntem Umgebung. Weiter Informationen findet man in [Montemerlo03].

---

<sup>1</sup>engl. *simultaneous localization and mapping*

## Literaturverzeichnis

- [Bouguet07] Jean-Yves Bouguet: *Camera Calibration Toolbox for Matlab*, Techn. Ber., MRL - Intel Corp., 2007, <http://robots.stanford.edu/cs223b04/JeanYvesCalib/>.
- [DCAM07] 1394 Trade Association: *DCAM*, Website, 2007, <http://www.1394ta.org/Technology/specifications/specifications.htm>.
- [Doucet98] Arnaud Doucet, Simon Godsill und Christophe Andrieu: *On sequential Monte Carlo sampling methods for Bayesian filtering*, Techn. Ber., Cambridge University Department of Engineering, 1998.  
URL <http://citeseer.ist.psu.edu/doucet00sequential.html>
- [Guppy07] Allied Vision Technology: *Technical Manual: AVT Guppy F-033B/C*, 2007, <http://www.alliedvisiontec.de/downloads/avt-cameras/avt-guppy.html>.
- [Hartley03] Richard Hartley und Andrew Zisserman: *Multiple view geometry in computer vision*, Cambridge University Press, Cambridge, UK, zweite Aufl., 2003, ISBN 0-521-54051-8.
- [ImgScr07] The Imaging Source: *Die Funktionsweise der Farbkamera*, pdf, 2007, <http://www.theimagingsource.com/de/resources/whitepapers/download/howcolcamswp.de.pdf>.
- [Intgeb07] Wikipedia: *Inkrementalgeber*, Website, 2007, <http://de.wikipedia.org/wiki/Inkrementalgeber>.
- [Kue07] Kübler: *Inkrementale Drehgeber*, pdf, 2007, [http://www.kuebler.com/PDFs/leaflet/drehgeber/deutsch/5802\\_d.pdf](http://www.kuebler.com/PDFs/leaflet/drehgeber/deutsch/5802_d.pdf).
- [Mel07] Melexis: *Angular Rate Sensor*, Website, 2007, [http://www.melexis.com/Sensor\\_ICs\\_Inertia/General/MLX90609\\_582.aspx](http://www.melexis.com/Sensor_ICs_Inertia/General/MLX90609_582.aspx).
- [Montemerlo03] Michael Montemerlo und Sebastian Thrun: *FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges*, *International Joint Conference on Artificial Intelligence*, 2003.

- [Munkres57] J. Munkres: *Algorithms For Assignment And Transportation Problems*, *J. Soc. Indust. Applied Mathematics*, Bd. 5:S. 32–38, March 1957.
- [Pilgrim07] R. A. Pilgrim: *Munkres Assignment Algorithm*, Website, 2007, <http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html>.
- [Richter05] Eric Richter: *Untersuchungen zur Lokalisierung von Fahrzeugen in Straßenszenen mittels Multi-Sensor-Daten-Fusion*, Diplomarbeit, Fakultät für Elektrotechnik und Informationstechnik, Lehrstuhl Nachrichtentechnik, 2005.
- [Scheunert06] Dr. Ullrich Scheunert: *Nachrichtentechnik 2: Schätztheorie und Multi-Sensor-Daten-Fusion*, Vorlesung, 2006, Fakultät für Elektrotechnik und Informationstechnik.
- [Thrun05] Sebastian Thrun, Wolfram Burgard und Dieter Fox: *Probabilistic Robotics*, The MIT Press, 2005, ISBN 9780262201629.
- [Wanka04] Prof. Dr. Gert Wanka: *Mathematik II*, Vorlesung für Ingenieure, 2004, Fakultät für Elektrotechnik und Informationstechnik.