

ALBERT-LUDWIGS-UNIVERSITÄT  
FREIBURG  
INSTITUT FÜR INFORMATIK

Arbeitsgruppe Autonome Intelligente Systeme  
Prof. Dr. Wolfram Burgard



Lernen von Linienmodellen aus Laserscannerdaten  
für mobile Roboter

Diplomarbeit

Daniel Sack

März 2003 – September 2003



# Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Außerdem erkläre ich, dass die Diplomarbeit nicht, auch nicht auszugsweise, bereits für andere Prüfungen angefertigt wurde.

(Daniel Sack)

Freiburg, den 08. September 2003



## Danksagung

An dieser Stelle bedanke ich mich bei den Menschen, die mich bei der Erstellung der Diplomarbeit in vielfältiger Weise unterstützt haben. Bei Herrn Prof. Dr. Wolfram Burgard bedanke ich mich für die ideenreiche Unterstützung und für die aufgebrauchte Zeit der angenehmen Betreuung.

Herzlicher Dank gebührt allen Mitarbeitern der Arbeitsgruppe Autonome Intelligente Systeme (Cyrill Stachniss, Rudolph Triebel, Maren Bennewitz, Dirk Hähnel, Dirk Zitterell und Patrick Pfaff), ohne deren Mithilfe diese Arbeit kaum möglich gewesen wäre. Sie standen mir bei Fragen jeder Zeit hilfreich zur Seite und haben eine gute Arbeitsatmosphäre geschaffen.

Auch meinem Mitbewohner Martin Rück und Till Soltész danke ich für ihre Verbesserungsvorschläge während der schriftlichen Ausarbeitung und ihrem Interesse an dieser Arbeit.

Mein besonderer Dank gebührt meiner ganzen Familie, die mich - nicht nur während des Studiums - finanziell und seelisch unterstützt haben.

Euch allen verdanke ich es, dass ich hier und Heute diese Arbeit abliefern kann.  
DANKE!



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>11</b>
1.1	Zielsetzungen .....	12
1.2	Aufbau der Arbeit .....	13
1.3	Verwandte Arbeiten .....	14
<b>2</b>	<b>Vermessung der Umgebung</b>	<b>17</b>
2.1	Sensorik .....	17
2.1.1	Lasersensor .....	17
2.1.2	Bestimmung der lokalen Ausrichtung der Messpunkte .....	18
2.1.3	Filterung von Scandaten .....	22
2.2	Umgebungsrepräsentation .....	23
2.2.1	Darstellung der Punkte, Geraden und Liniensegmente .....	23
2.2.2	Selbstlokalisierung .....	24
2.2.3	Filterung von dynamischen Messpunkten .....	25
<b>3</b>	<b>Ein heuristisches Verfahren</b>	<b>29</b>
3.1	Geradenextraktion mittels der Hough-Transformation .....	29
3.1.1	Grundidee der Hough-Transformation .....	30
3.1.2	Der Hough-Algorithmus .....	31
3.1.3	Erweiterung durch die lokale Ausrichtung .....	33
3.1.4	Extraktion von Geraden .....	34
3.2	Übergang von einer Geraden zu einem Liniensegment .....	35
3.2.1	Zuordnung von Punkten zu einer Geraden .....	35
3.2.2	Segmentierung einer Geraden .....	36
3.2.3	Filterung der Liniensegmente .....	38

3.3	Anpassung eines Liniensegments an eine Punktmenge .....	40
3.3.1	Line-fitting als lineare Regression .....	40
3.3.2	Line-fitting als nicht-lineare Regression.....	42
3.4	Abbruchkriterium.....	43
3.5	Bearbeitung der Liniensegmente .....	44
3.5.1	Vereinigung von Liniensegmenten .....	44
3.5.2	Veränderung der Länge der Liniensegmente .....	48
3.5.3	Einfügen „kurzer“ Liniensegmente.....	51
3.6	Flussdiagramm des Verfahrens .....	53
<b>4</b>	<b>Lernen von Linienmodellen mit EM</b> .....	<b>55</b>
4.1	Geraden- und Punktmodell .....	56
4.2	Die Wahrscheinlichkeitsfunktion.....	57
4.3	Erwartungsmaximierung.....	58
4.3.1	Der E-Schritt.....	59
4.3.2	Der M-Schritt.....	61
4.4	Bestimmung der Anzahl von Geraden.....	62
4.5	Abbruchkriterium.....	66
4.6	Übergang von Geraden zu Liniensegmenten .....	68
4.7	Bearbeitung der Liniensegmente .....	71
4.7.1	Löschen von Liniensegmenten .....	71
4.7.2	Veränderung der Länge der Liniensegmente.....	73
4.8	Flussdiagramm des Verfahrens .....	74
<b>5</b>	<b>Experimente</b> .....	<b>75</b>
5.1	Beurteilungskriterien.....	75
5.2	Einfluss der Parameter .....	76
5.2.1	Heuristisches Verfahren.....	76
5.2.2	EM-Verfahren.....	82
5.3	Einfluss der lokalen Ausrichtung.....	88
5.3.1	Heuristisches Verfahren.....	88
5.3.2	EM-Verfahren.....	92
5.4	Vergleich der beiden Verfahren .....	100



<b>6</b>	<b>Zusammenfassung</b>	<b>107</b>
6.1	Ausblick .....	108
<b>A</b>	<b>Parameter</b>	<b>111</b>
A.1	Vermessung der Umwelt .....	111
A.2	Heuristisches Verfahren.....	112
A.3	EM-Verfahren.....	113
	<b>Literaturverzeichnis</b>	<b>115</b>



# 1 Einleitung

Mobile Roboter haben sich in den letzten Jahren immer mehr zu autonomen Systemen weiterentwickelt. Es ist immer schon der Traum vieler Menschen gewesen, intelligente und selbstständig handelnde Systeme zu entwickeln, welche als zusätzliche Arbeitskraft für verschiedene Aufgaben eingesetzt werden können. Damit eine permanente Kontrolle durch Menschen überflüssig wird, müssen Roboter möglichst selbständig in ihrer eingesetzten Umgebung arbeiten. Die Frage der Umgebungserkennung und Sensorinterpretation ist ein wesentlicher Aspekt jeder Anwendung autonomer mobiler Roboter. Nach Russel und Norvig [1994] sind für einen reflexiven Agenten bzw. Roboter die momentanen Sensordaten vollkommen ausreichend, um seine nächste Aktion zu bestimmen. Auch wenn sich diese Art der Kontrolle durch schnelle Reaktionen auszeichnet, so wird sie in komplexen Situationen bald an ihre Grenzen stoßen. Für eine zielgerichtete, geplante Aktion werden gespeicherte Informationen und Annahmen über den Zustand der Umwelt benötigt. Eine Sequenz von Wahrnehmungen des Roboters wird zu einem sogenannten Weltmodell zusammengefasst, um ein genaueres Verständnis über die Umgebung zu erhalten. Erst mit der Existenz eines Weltmodells ist der Roboter in der Lage, seine Aktionen zu planen und kontrolliert auszuführen. Um beispielsweise einen Roboter von seiner Position zu einem bestimmten Ort zu bewegen, benötigt er seine momentane Position und eine Vorstellung über die Struktur der Umgebung. Anhand eines Weltmodells der Umgebung kann sich der Roboter dann selbst lokalisieren und eine Pfadplanung durchführen. Ohne ein Weltmodell bliebe als Alternative eine rein zufällige und ineffiziente Suche.

Es gibt bereits mehrere Robotersysteme, die ihre Umgebung durch verschieden Modelle repräsentieren und hervorragend in ihrer Umwelt navigieren können. Ein Beispiel dafür ist der interaktive Museumsführer Rhino [Burgard et al. 1998], welcher im Deutschen Museum in Bonn eingesetzt wurde. Bei dieser Anwendung wurde die Umgebungskarte allerdings nicht selbstständig vom Roboter erzeugt, sondern von Hand mittels eines Maßbandes erstellt. Eine solche Vermessung eines ganzen Gebäudes ist sehr aufwendig. Für den Roboter stellt sich daher die Aufgabe, von einer unbekanntem Umgebung eine für ihn nutzbare Karte selbstständig zu erstellen. Durch eine autonome Explorationsfahrt in alle Bereiche einer Umgebung werden mit verschiedenen Sensoren Informationen gesammelt, um daraus eine möglichst genaue Karte zu erstellen.

Heutzutage können bereits komplette dreidimensionale Modelle von Umgebungen erstellt werden (z.B. [Hähnel et al. 2001]) und man könnte vermuten, das Problem der zweidimensionalen Kartenerstellung sei schon längst gelöst. In der Praxis ergeben sich jedoch einige Probleme, wie wir sehen werden. Die Erstellung exakter Modelle ist ein sehr anspruchsvolles und noch nicht vollständig gelöstes Forschungsgebiet in der AMR<sup>1</sup>-Entwicklung.

In dieser Arbeit möchten wir zwei verschiedene Ansätze zur Erstellung von metrischen Weltmodellen vorstellen. Ausgehend von Sensordaten, die zu sogenannten Punktemodellen zusammengefasst werden, erzeugen wir Linienmodelle, welche aus Mengen von Liniensegmenten bestehen. Der Vorteil der auf geometrischen Objekten, bei uns sind es Punkte und Linien, basierenden Modelle ist die hohe Genauigkeit, die nur durch die benützte Zahlendarstellung beschränkt ist. Die Reduzierung auf Linien bei der Beschreibung der Umgebung ergibt eine effiziente, kompakte und gut visualisierbare Darstellungsform. Grundsätzlich gilt jedoch, dass ein höherer Abstraktionsgrad in der Darstellung auch ein aufwendigeres Analyseverfahren zur Erzeugung der Modelle bedingt.

### 1.1 Zielsetzungen

Das Ziel dieser Arbeit ist das Erstellen von Linienmodellen aus Laserscannerdaten für mobile Roboter. Ein Roboter vermisst mittels eines Lasersensors seine Umgebung und erstellt aus den erhaltenen Messdaten zunächst ein Punktemodell. Da sich bei der Vermessung dynamische Hindernisse, wie zum Beispiel Menschen, in der Umgebung befinden können, versuchen wir diese Hindernisse aus dem Punktemodell herauszufiltern. Die meisten Einsatzgebiete der Roboter weisen in der Regel einen gewissen Grad der Strukturiertheit auf, der bei Innenraumumgebungen sowohl von Gebäudemerkmale als auch von Einrichtungsgegenständen herrührt. Dadurch ergibt sich die Möglichkeit, bei den Weltmodellen geometrische Primitive wie Liniensegmente zu benützen. Der Hauptteil dieser Arbeit beschäftigt sich mit der Umwandlung von Punktemodellen in Linienmodelle. Wir möchten erreichen, dass ein Linienmodell alle linearen Strukturen einer Umgebung möglichst exakt beschreibt. Bei der Erstellung extrahieren wir anhand der Menge von Punkten eines Punktemodells Liniensegmente und fassen diese zu einem Linienmodell zusammen. Das Ersetzen der Messpunkte durch Liniensegmente ist eine Form der Kartenkompression und wir können eine kompaktere, effiziente und trotzdem genaue Beschreibung der Umgebung erhalten. Bereits bei der Vermessung einer Umgebung, aus der ein Punktemodell entsteht, versuchen wir den Messpunkten Informationen über mögliche lineare

---

<sup>1</sup> AMR = Autonome Mobile Roboter

Strukturen zuzuordnen. Der Clou an unseren Verfahren ist, dass wir diese Informationen bei der Extraktion der Liniensegmente zusätzlich verwenden können und nicht nur allein auf die Information über die Lage der Messpunkte angewiesen sind. Dadurch versprechen wir uns in einigen Situationen verbesserte Linienmodelle. Wie dies im Einzelnen in die Praxis umgesetzt werden kann, beschreiben wir in den folgenden Kapiteln.

## 1.2 Aufbau der Arbeit

Dieses einleitende Kapitel wird durch die Vorstellung verwandter Arbeiten abgeschlossen. Das verschafft uns zunächst einen Überblick über die verwendeten Thematiken.

Im zweiten Kapitel gehen wir auf die Vermessung der Umgebung mittels Lasersensoren ein. Den erzeugten Messpunkten werden zusätzliche Informationen über mögliche lineare Strukturen zugeordnet und werden zu einem Punktemodell der Umgebung zusammengefasst, welches zumeist nur lineare und statische Hindernisse darstellt.

In Kapitel 3 wird ein heuristisches Verfahren zur Erstellung von Linienmodellen vorgestellt, welches auf der Hough-Transformation basiert. Wir beschreiben die einzelnen Schritte dieses Verfahrens, das anhand eines Punktemodells aus Kapitel 2 nacheinander Liniensegmente extrahiert und diese zu einem Linienmodell zusammenfasst.

Eine zweite Methode zur Erstellung von Linienmodellen wird in Kapitel 4 beschrieben, welche auf einer Erwartungsmaximierung (EM) beruht. Wir wollen dabei Punktemodelle durch Geradenmodelle mit einer möglichst hohen erwarteten Wahrscheinlichkeit erklären. Anschließend werden die Geraden in Liniensegmente umgewandelt, die dann unsere gewünschten Linienmodelle bilden.

Das fünfte Kapitel enthält eine Auswahl von Experimenten, die in realen Umgebungen durchgeführt wurden. Wir zeigen die Leistungsfähigkeit verschiedener Ansätze der zwei vorgestellten Verfahren und vergleichen beide auf ihre Anwendungsmöglichkeiten in verschiedenen räumlichen Umgebungen.

Das letzte Kapitel enthält eine Zusammenfassung dieser Arbeit und gibt einen Ausblick auf weiterführende Ansätze und mögliche Erweiterungen.

### 1.3 Verwandte Arbeiten

Auf dem Gebiet der Umgebungsmodellierung aus Laserscannerdaten gibt es bereits zahlreiche Arbeiten und verschiedene Ansätze. Wir werden in diesem Abschnitt die für unsere Aufgabenstellung benötigten Forschungsbereiche kurz erklären und anhand bisheriger Arbeiten Gemeinsamkeiten sowie Unterschiede zu dieser Arbeit aufzeigen.

Eine Umgebungskarte kann durch eine Explorationsfahrt des Roboters und den dadurch gewonnenen Sensordaten erstellt werden. Das größte Problem bei der Exploration der Umgebung ist normalerweise die Positionsbestimmung des Roboters, die für eine korrekte Eintragung der detektierten Umgebungsmerkmale in eine Karte benötigt wird. Für die Lösung dieses Problems wurden in der Vergangenheit zahlreiche Methoden entwickelt. In diese Arbeit wird ein bewährtes Verfahren von Hähnel et al. [2003] verwendet. Wir beschäftigen uns weniger mit der Positionsbestimmung, da wir uns auf die Interpretation der Messdaten konzentrieren wollen.

Bei der Vermessung einer Umgebung wird versucht, mittels verschiedene Verfahren Merkmale direkt aus einem einzelnen Scan zu extrahieren, wie zum Beispiel Linien und Ecken. Bei der weiteren Verarbeitung der Daten ist es oftmals ein Vorteil, nicht die einzelnen Scanpunkte zu verwenden, sondern mit den extrahierten Merkmalen zu arbeiten. Zum Beispiel bei Crowley [1989], Arras und Siegwart [1997], Pfister et al. [2003] und Gonzalez et al. [1994] wird anhand von Merkmalen ein Modell der Umgebung entwickelt. In dieser Arbeit verzichten wir bei der Erstellung eines Linienmodells auf den direkten Übergang von Messpunkten eines Scans zu Merkmalen. Wir vermessen zuerst die komplette Umgebung und sammeln genügend Messpunkte, welche zu einem Punktemodell der Umgebung zusammengefasst werden. Die Extraktion von Linien wird erst auf das Punktemodell angewendet und nicht schon auf die einzelnen Scans. Wir führen aber auch eine Linienextraktion aus den einzelnen Scans durch, jedoch ersetzen wir nicht die Messpunkte durch Linien, sondern ordnen den Messpunkten Informationen über diese Linien zu. Diese zusätzlichen Informationen können bei der Erstellung eines Linienmodells verwendet werden; dadurch vernachlässigen wir nicht komplett die Vorteile der extrahierten Linien aus einem einzelnen Scan.

Bei der Verarbeitung der erstellten Messdaten zu einer Karte bilden Rasterkarten mit Belegungswahrscheinlichkeiten eine sehr populäre Klasse der Umgebungsbeschreibung [Moravec und Elfes 1985]. Diese Karten beschreiben nicht die charakteristischen geometrischen Merkmale von Hindernissen in der Umgebung und scheinen deswegen nicht für die Aufgabenstellung diese Arbeit geeignet. Andererseits eignen sich diese Karten sehr gut für die Erkennung und Verfolgung von dynamischen Objekten und werden in der Praxis häufig eingesetzt [Wang und Thorpe 2002]. Da ein Modell einer Umgebung nur statische Hindernisse beschreiben soll, bilden wir vor der eigentlichen

Erstellung unseres Linienmodells eine Rasterkarte mit Reflektionswahrscheinlichkeiten, um dynamische Messpunkte aus unseren Daten herauszufiltern. Wir orientieren uns an der Arbeit von Thrun et al. [2003], die wir für unsere Situation an einigen Stellen etwas vereinfachen.

Für die Extraktion von Linien aus einer Menge von Punkte gibt es mehrere erfolgreiche Methoden. Ein Verfahren, das in der Robotik häufig als *split-and-merge*-Verfahren bezeichnet wird, wird auf verschiedene Weisen unter anderem von Grandjean und de Saint Vincent [1989], Baltzakis und Trahanias [2002], Newman et al. [2002] und Gutmann [2000] eingesetzt und führt zu guten Ergebnissen. Dieses Verfahren kann nur auf einen einzelnen Scan angewendet werden, da die Reihenfolge der Erstellung der Messpunkte benötigt wird. Wir verwenden das *split-and-merge*-Verfahren in dieser Arbeit bei der Linienextraktion aus einem einzelnen Scan und passen es an unsere Aufgabe an.

Ein weiterer Ansatz zur Linienextraktion beruht auf der Hough-Transformation und wird bei der Kartenerstellung zahlreich verwendet, zum Beispiel von Arras und Siegwart [1997], Pfister et al. [2003] und Leonard et al. [2001]. Für die Linienextraktion aus mehreren Scans bildet die Hough-Transformation die Grundlage für ein in dieser Arbeit vorgestelltes Verfahren. Nach einer Extraktion versuchen wir die Lage eines Liniensegments zu verbessern, indem ein Segment an eine zugeordnete Punktemenge durch ein *line-fitting* optimal angepasst wird.

Um lineare Strukturen in der Umgebung zu finden, verwenden wir in dem anderen vorgestellten Verfahren dieser Arbeit eine Variante der Erwartungsmaximierung (EM). In bisherigen Arbeiten ist dieser Ansatz weniger für die Erstellung von 2D-Modellen verwendet worden, da bei der Erstellung eines Modells die gesamte Datenmenge von Messpunkten vielfach durchlaufen werden muss. Dadurch besitzen EM-Algorithmen zumeist eine verhältnismäßig große Laufzeit im Vergleich zu den anderen Linienextraktionsverfahren. Es gibt aber auch Arbeiten in denen 3D-Modell aus Laserscannerdaten mit einem EM-Algorithmus erstellt werden, wie zum Beispiel von Martin und Thrun [2002] und Burgard et al. [2001]. Diese Arbeiten haben uns dazu veranlasst, einen EM-Algorithmus zur Erstellung von Linienmodellen zu entwickeln, da der Unterschied zu unserer Situation nur sehr gering ist. Anstatt in einer dreidimensionalen Umgebung Ebenen an Messpunkte anzupassen, wird unser Verfahren in einer zweidimensionalen Umgebung eingesetzt und verwendet Geraden anstelle von Ebenen.





## 2 Vermessung der Umgebung

Bevor ein komplettes Weltmodell erstellt werden kann, muss die Umgebung des Roboters zuerst einmal vermessen werden. Während der Vermessung bzw. anschließend kann dann ein Weltmodell erstellt werden, das dem Roboter eine bessere Planung seiner Aktionen in der Umgebung ermöglicht.

Dieses Kapitel beschäftigt sich mit der Gewinnung von Messpunkten mittels Sensoren und der Zusammenfassung aller Messpunkte zu einem vorläufigen Weltmodell. Dieses Weltmodell bildet die Grundlage für die in Kapitel 3 und 4 vorgestellten Verfahren.

### 2.1 Sensorik

Für ein mit seiner Umgebung interagierendes System sind die Sinnesorgane unverzichtbare Bestandteile. Wie in der Natur gibt es auch in der Robotik verschiedene Sensoren. Dabei ist die Wahl des geeigneten Sensors von den Aufgaben und Einsatzbedingungen des Roboters abhängig. Bei der Vermessung einer Umgebung werden vor allem entfernungsmessende Sensoren benötigt. In dieser Arbeit wurden ausschließlich Lasersensoren verwendet, die eine Umgebung mittels eines Laserstrahls abtasten, indem sie ein Signal senden und auf das zurückkehrende „Echo“ warten. Die Sensoren messen die Zeit zwischen Aussenden und Empfangen des Signals und bestimmen somit die Entfernung eines Hindernisses. Im folgenden Abschnitt wird die Verwendung von Lasersensoren genauer beschrieben. In den *Abschnitten 2.1.2* und *2.1.3* wird auf den Lasersensoren aufbauend erläutert, wie wir weitere Informationen aus den Messdaten erhalten können, die bei der späteren Erstellung eines Linienmodells gezielt eingesetzt werden können.

#### 2.1.1 Lasersensor

Rotierende Lasersensoren wie SICK-PLS<sup>2</sup> haben sich als Standardsensoren für autonome mobile Roboter etabliert. Diese 2D-Lasersensoren tasten die Umgebung in

---

<sup>2</sup> Laserscanner vom Typ PLS (Proximity Laser Scanner) der Firma SICK

einer horizontalen Rotationsebene ab, wodurch der Roboter in einem Gebäude eine Art Grundriss der Umgebung erzeugt. Die von uns verwendeten Sensoren besitzen einen Öffnungswinkel von  $180^\circ$  und liefern pro Umdrehung, auch Scan genannt, eine Menge von Messwerten  $\{m_i = (r_i, \varphi_i) | i = 0, \dots, n\}$ , die wir als Scanpunkte bezeichnen. Bei der Winkelauflösung des Sensors kann zwischen  $1/2^\circ$  und  $1^\circ$  gewählt werden. Dies ergibt  $n = 181$  bzw.  $n = 361$  Messwerte pro Scan, die ca. alle 250 ms erstellt werden können. Ein Lasersensor kann bis zu 50 m entfernte Hindernisse mit einer Distanzauflösung von ca. 1-2 cm wahrnehmen.

Ein Scanpunkt  $m_i = (r_i, \varphi_i)$  kann mit der gegebenen Aufnahmeposition  $(x_r, y_r)$  des Roboters und dessen Orientierung  $\theta_r$  wie folgt in einen Punkt  $z_i$  mit den absoluten kartesischen Koordinaten  $x_i$  und  $y_i$  umgerechnet werden:

$$\vec{z}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} x_r \\ y_r \end{pmatrix} + r_i \cdot \begin{pmatrix} \cos(\theta_r + \Delta(\varphi_i, \theta_r)) \\ \sin(\theta_r + \Delta(\varphi_i, \theta_r)) \end{pmatrix}$$

mit  $\Delta(\varphi_i, \theta_r)$  als Winkel des Laserstrahl  $i$  von  $\theta_r$  aus gesehen. Die Bestimmung der Aufnahmeposition des Roboters wird in *Abschnitt 2.2.2* beschrieben. Ein solches  $z_i$  wird als Hindernispunkt bzw. Messpunkt in unserer Umgebung bezeichnet.

Die Messungen eines Scans erfolgen für gewöhnlich in einer festen Reihe, beispielsweise mit aufsteigendem Winkel. Diese Eigenschaft kann bei der Vorverarbeitung von Messpunkten eines Scans ausgenutzt werden.

### 2.1.2 Bestimmung der lokalen Ausrichtung der Messpunkte

Oftmals ist es von Vorteil, nicht direkt mit den einzelnen Messpunkten zu arbeiten, sondern Merkmale aus einem kompletten Scan zu extrahieren. In der Robotik extrahieren einige Verfahren Liniensegmente aus einem Scan und verwenden nur noch diese für die Erstellung eines Linienmodells (z.B. [Gutmann et al. 2002], [Pfister et al. 2003] und [Arras et al. 1996]). Wir versuchen hingegen in dieser Arbeit zuerst so viele Messpunkte wie möglich von unserer Umgebung zu sammeln, um erst dann Liniensegmente aus allen Messpunkten zu erstellen. Um trotzdem die Merkmale aus einem einzelnen Scan zu berücksichtigen, versuchen wir den einzelnen Messpunkten zusätzliche Informationen zuzuordnen. Diese Informationen erhalten wir aus einem einzelnen Scan und können uns bei der späteren Berechnung von Liniensegmenten weiterhelfen.

Für das Erstellen von Liniensegmenten wird nun ein Algorithmus vorgestellt, der aus den Punkten eines Scans Geraden bildet. Jedem einzelnen Messpunkte wird genau eine Ausrichtung einer Geraden als Merkmal zugeordnet. Ein solches Merkmal bezeichnen wir in dieser Arbeit als *lokale Ausrichtung* eines Punktes. Das Merkmal wird als lokal bezeichnet, da wir diese Information über die Umgebung aus einem einzelnen Scan

erhalten, der die Umgebung von genau einer Position aus beschreibt. Die Ausrichtung  $\psi$  einer Geraden mit  $-\frac{\pi}{2} < \psi \leq \frac{\pi}{2}$  wird als Winkel zwischen der positiven x-Achse und der Geraden definiert. Die Zuordnung der lokalen Ausrichtung für jeden Messpunkt ist eine Neuheit im Vergleich zu bisherigen Arbeiten.

Linienerkennung aus 2D Laserscannerdaten wird recht erfolgreich in der Robotik eingesetzt und kann, je nach gegebener Situation, auf verschiedene Weisen durchgeführt werden. In vielen Algorithmen wird die Hough-Transformation [Hough 1962] als Grundlage zur Linienerkennung benutzt, wie auch von Pfister et al. [2003], Arras und Siegwart [1997] und Leonard et al. [2001]. Die Hough-Transformation ist in der Praxis sehr beliebt, aber wir werden sie erst in unserem heuristischen Verfahren in *Abschnitt 3.1* verwenden, da für unsere jetzige Situation sich eher Verfahren eignen, welche die Reihenfolge der Messpunkte bei der Vermessung berücksichtigen.

Wir verwenden für die Bestimmung der lokalen Ausrichtung eine Variante des *split-and-merge*-Algorithmus. Jeder Messpunkt wird einer Geraden zugeordnet und erhält somit seine lokale Ausrichtung. Mit der *split&merge*-Funktion aus *Algorithmus 1* werden anhand der Menge  $Z_{0,n} = \{z_i \mid i = 0, \dots, n\}$  von Messpunkten eines kompletten Scans rekursiv Geraden berechnet. Dabei ist zu beachten, dass die Menge  $Z_{0,n}$  nach der Reihenfolge der Erstellung geordnet ist.

---

**Algorithmus 1:** *split & merge*( $Z_{i,j}$ )

**Eingabe:** Menge  $Z_{i,j} = \{z_i, \dots, z_j\}$  von geordneten Messpunkten

**Ausgabe:** Menge  $Z^*$  von Messpunkten mit lokaler Ausrichtung

**Ablauf:**  $Z^* = \emptyset$

$g = \text{optimale\_Gerade}(Z_{i,j})$

**if**  $\max\_Abstand(Z_{i,j}, g) \leq \text{SPLIT\_DIST\_MAX}$  **then**

$Z_{i,j}^* = Z_{i,j}$  mit lokaler Ausrichtung von  $g$

$Z^* = Z^* \cup Z_{i,j}^*$

**else**

$g' = \text{Gerade}(z_i, z_j)$

$splt = \text{Index des Punktes aus } Z_{i,j} \text{ mit max. Abstand zu } g'$

$Z^* = Z^* \cup \text{split \& merge}(Z_{i,splt}) \cup \text{split \& merge}(Z_{splt,j})$

**endif**

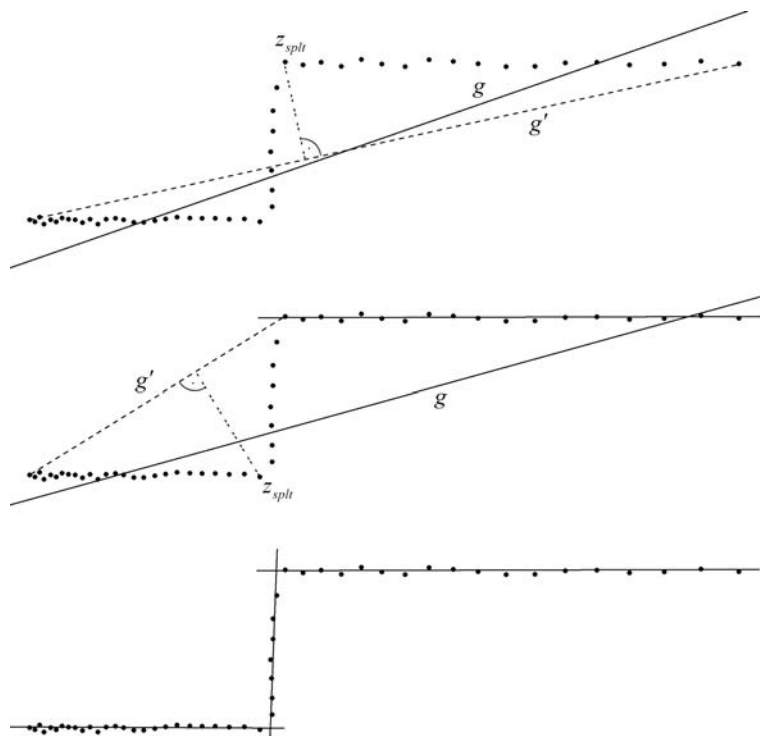
**return**  $Z^*$

---

Zu Beginn der *split&merge*-Funktion wird eine Gerade  $g$  erzeugt, welche an die komplette Punktemenge  $Z_{i,j}$  angepasst wird. Die Funktion *optimale\_Gerade*( $Z_{i,j}$ ) erzeugt diese Gerade mit minimalem Abstand zu den Messpunkten. Wie eine solche Gerade erzeugt werden kann, wird in *Abschnitt 3.3* erklärt. Danach wird geprüft, ob

der maximale Abstand eines Punktes aus  $Z_{i,j}$  zur Geraden  $g$  kleiner als  $SPLIT\_DIST\_MAX^3$  ist. Ist dies der Fall, dann wird jedem Messpunkt aus  $Z_{i,j}$  die lokale Ausrichtung zugeordnet. Die lokale Ausrichtung wird gleich der Ausrichtung der Geraden gesetzt. Anschließend werden die Punkte in die Ausgabemenge  $Z^*$  aufgenommen. Die *split&merge*-Funktion ist in diesem Fall beendet. Ist der maximale Abstand jedoch größer als  $SPLIT\_DIST\_MAX$ , dann wird die Punktemenge an einer bestimmten Stelle in zwei Mengen aufgeteilt. Die *split&merge*-Funktion wird anschließend rekursiv mit den beiden Teilmengen aufgerufen. Die Vereinigung der beiden von der Funktion bestimmten Punktemengen bildet dann die Ausgabemenge  $Z^*$ . Um die Stelle für die Aufteilung zu finden, wird der Messpunkt gesucht, welcher am weitesten von der Geraden  $g'$  entfernt ist, wobei  $g'$  durch die zwei Punkte  $z_i$  und  $z_j$  verläuft.

Um diesen Algorithmus zu verdeutlichen, zeigt *Abbildung 2.1* ein Beispiel, in dem den Messpunkten eines simulierten Scans Geraden zugeordnet werden.



**Abbildung 2.1:** *Split-and-merge*-Algorithmus, bei dem den Messpunkten eines Scans optimal angepasste Geraden zugeordnet werden

---

<sup>3</sup> Alle in diesen Kapitel benötigten Parameter sind im Anhang zusammenfassend aufgelistet und werden dort kurz beschrieben (Seite 111).

In diesem Algorithmus wird der Parameter  $SPLIT\_DIST\_MAX$  benötigt, der für die Anzahl verschiedener lokaler Ausrichtungen und Qualität der lokalen Ausrichtungen verantwortlich ist. Der Parameter legt den maximalen Abstand fest, den ein Messpunkt zu einer Geraden besitzen darf, damit diese Gerade die lokale Ausrichtung des Messpunktes festlegt. Kleinere Werte für  $SPLIT\_DIST\_MAX$  erzeugen im Mittel mehr unterschiedliche Geraden als größere Werte, wodurch die Genauigkeit der Geraden bei den kleineren Werten besser wird. Zu klein darf der Wert jedoch nicht gewählt werden, da ansonsten im schlechtesten Fall immer nur zwei benachbarte Punkte die Lage der Geraden bestimmen. In dieser Arbeit wurde der Parameter  $SPLIT\_DIST\_MAX$  mit einem Wert von 5 cm belegt, wodurch wir in den Experimenten für unsere Situation akzeptable Ergebnisse erzielen konnten.

Unser Algorithmus basiert auf einem typischen *divide-and-conquer*-Ansatz. Die Laufzeit ist mit  $n$  als Anzahl der Messpunkte im schlechtesten Fall  $O(n^2)$  und im mittleren Fall  $O(n \log n)$ .

Die Grundidee dieses Algorithmus ist vergleichbar mit dem *iterativ-end-point-fit*-Verfahren, das erstmals von Duda und Hart [1973] vorgestellt wurde. In der Mathematik hingegen ist dieser Algorithmus ähnlich der Polynomvereinfachung von Douglas und Peucker [1973], bei der anhand einer Punktemenge eines Polynoms ein Polygon erstellt wird. Die Laufzeit dieses Verfahrens beträgt wie in unserem Verfahren im schlechtesten Fall  $O(n^2)$ , bei einer Anzahl von  $n$  Punkten. Diese Laufzeit wurde später durch Hershberger und Snoeyink [1992] auf  $O(n \log n)$  verbessert. Der wesentliche Unterschied zu unserem Verfahren ist, dass die erzeugten Geraden bzw. Liniensegmente nicht an Punktemengen optimal angepasst werden, sondern durch jeweils zwei Punkte definiert werden. Bei unserem Algorithmus werden diese Geraden nur bei der Aufteilung der Punktemenge gebildet. Die zugeordneten Geraden, die für die lokale Ausrichtung verantwortlich sind, werden an Punktemengen optimal angepasst. Durch die Anpassung wird die Lage einer Geraden nicht nur durch zwei Punkte festgelegt, sondern auch durch alle Messpunkte die zwischen den zwei Punkten liegen. Da all diesen Punkten die gleiche Gerade zugeordnet wird, sollte die Lage der Geraden auch von allen Punkten abhängen.

Ein Messpunkt  $z_i$  wird nach der Berechnung der lokalen Ausrichtung nun neuerlich durch drei Werte beschrieben:

$$z_i = (x_i, y_i, \omega_i)$$

mit  $(x_i, y_i)$  als kartesische Koordinaten und zusätzlich  $\omega_i$  als lokale Ausrichtung. Im Folgenden wird die lokale Ausrichtung der Messwerte an vielen Stellen dieser Arbeit verwendet. Durch die lokale Ausrichtung erhalten wir die Information, wie das Hindernis eines Messpunktes in der Umgebung ungefähr ausgerichtet ist. Ein Messpunkt enthält somit nicht nur die Information über die Lage eines gesannten Hindernisses, sondern auch noch die tendenzielle Ausrichtung. Im *Abschnitt 5.3* wird

anhand von Experimenten verglichen, ob diese zusätzliche Information unsere erstellten Linienmodelle im Endeffekt auch wirklich verbessert.

### 2.1.3 Filterung von Scandaten

Ein Problem der entfernungsmessenden Sensoren liegt darin, dass der von ihnen ausgesendete Impuls nicht immer unmittelbar zu ihnen zurückgeworfen wird, sondern entweder am nächstliegenden Hindernis gespiegelt wird oder zum Beispiel eine Glasscheibe ungehindert durchdringt. Hierdurch erkennt der Sensor ein anderes Hindernis in größerer Entfernung und gibt somit implizit den Bereich des tatsächlichen Hindernisses fälschlich als frei an. Für die spätere Verarbeitung der Messwerte ist es daher von Vorteil, die Scanpunkte vorher zu filtern, um fehlerhafte Messwerte zu entfernen.

Ein weiteres Problem bei der Verwendung von Lasersensoren ist der zunehmende Fehler in den Messwerten, je weiter das Hindernis von Roboter entfernt ist. Das hängt vor allem mit der ungenauen Selbstlokalisierung (*Abschnitt 2.2.2*) zusammen. Um die Fehler in den Messwerten möglichst gering zu halten, verwenden wir deshalb nur Messwerte  $(r_i, \varphi_i)$  eines Scans, deren Abstände  $r_i$  vom Roboter kleiner gleich einem bestimmten Wert sind, der im Folgenden mit *SCAN\_DIST\_MAX* bezeichnet wird. Dadurch entsteht ein höherer Aufwand für eine lückenlose Vermessung der Umgebung, da der Roboter an jedem Hindernis nah genug vorbeifahren muss. Für die vermessenen Umgebungen dieser Arbeit liefern Werte zwischen 5m und 10m für *SCAN\_DIST\_MAX* ein gutes Punktemodell mit nur geringen Fehlern in den Messwerten.

Für die Erstellung unserer Linienmodelle haben wir vorausgesetzt, dass die Umgebungen zumeist aus linearen Strukturen bestehen. Alle nicht-linearen Strukturen sind nur schwer durch Liniensegmente darstellbar. Wir versuchen deshalb Messpunkte von nicht-linearen Hindernissen herauszufiltern. In den *Algorithmus 1* zur Bestimmung der lokalen Ausrichtung kann ein solcher Filter eingebaut werden. Die *split&merge*-Funktion bildet eine Gerade, die an eine Menge von Punkten angepasst wird. Ist die Punktemenge zu klein, dann gehören diese Punkte wahrscheinlich nicht zu einem linearen Hindernis und können entfernt werden. Wir führen den Parameter *SPLIT\_POINTS\_MIN* ein, der die minimale Anzahl von Messpunkten festlegt, für die eine optimale Gerade erstellt wird. Ist die Punktemenge kleiner als *SPLIT\_POINTS\_MIN*, dann wird keine optimale Gerade gebildet und alle Messwerte dieser Menge werden entfernt. Einzelne Ausreißer in den Messungen werden dadurch auch herausgefiltert, da durch Ausreißer meistens sehr kleinen Punktemengen entstehen, in denen sie enthalten sind.

Die Einführung des Parameters *SPLIT\_POINTS\_MIN* in die *split&merge*-Funktion hat noch einen weiteren positiven Effekt. Bei einer geringen Anzahl von Punkten haben

fehlerhafte Messpunkte sehr großen Einfluss auf die Lage einer optimalen Geraden, da die Gerade nur an wenig Messpunkte angepasst wird. Dadurch kann die lokale Ausrichtung der Messpunkte fehlerhaft werden. Durch *SPLIT\_POINTS\_MIN* wird festgelegt, dass die Gerade an nicht zu wenig Messwerten angepasst wird und dadurch wird eine fehlerhafte lokale Ausrichtungen in dem meisten Fällen vermieden. In der Praxis haben sich Werte für *SPLIT\_POINTS\_MIN* zwischen 5 und 10 bewährt.

## 2.2 Umgebungsrepräsentation

Für die Erstellung von Punkte-, Geraden- und Linienmodellen benutzen wir metrische Weltmodelle. Diese Klasse von Modellen enthält Datenstrukturen, welche die Position von Umgebungsmerkmalen bezüglich eines gegebenen Koordinatensystems speichern.

Im folgenden Abschnitt werden die verschiedenen Arten und Darstellungsformen der Umgebungsmerkmale vorgestellt, die in dieser Arbeit verwendet werden. Danach wird darauf eingegangen, wie wir unsere Messwerte in das Koordinatensystem korrekt einfügen können. Als Letztes wird dann noch eine Rasterkarte der Umgebung vorgestellt, die es uns ermöglicht, dynamische Merkmale aus unserem Modell zu entfernen.

### 2.2.1 Darstellung der Punkte, Geraden und Liniensegmente

In unseren Verfahren werden insgesamt drei verschiedene geometrische Strukturen verwendet. Diese Strukturen bzw. unsere Modelle werden in einem kartesischen Koordinatensystem dargestellt. In manchen Situationen steigen wir auf ein Polarkoordinatensystem um, welches uns die Berechnung einzelner Strukturen in einigen Situationen vereinfacht. Es werden nun die einzelnen Strukturen mit ihren verschiedenen Darstellungsformen vorgestellt:

#### **Punkte**

Ein Punkt  $p$  im kartesischen Koordinatensystem wird durch seine  $x$ - und  $y$ -Werte beschrieben.

$$p = (x, y)$$

Ein Messpunkt  $z$  besitzt zusätzlich eine lokale Ausrichtung  $\omega$ .

$$z = (x, y, \omega)$$

Ein Punkt  $p$  im kartesischen Koordinatensystem wird durch eine Sinuskurve im Polarkoordinatensystem beschrieben.

$$p: r = x \cdot \cos \varphi + y \cdot \sin \varphi$$

mit  $r \in \mathbb{R}$  und  $\varphi \in [0, \pi)$ .

### Geraden

Im kartesischen Koordinatensystem kann eine Gerade  $l$  auf viele verschiedene Weisen dargestellt werden. Wir verwenden die Punkt-Steigungs-Form

$$l: y = m \cdot x + b,$$

mit  $m$  als Steigung einer Geraden und  $b$  als Abschnitt auf der y-Achse. Aus  $m$  lässt sich wie folgt die Ausrichtung  $\psi$  einer Geraden berechnen, welche unter anderem für die Bestimmung der lokalen Ausrichtung eines Punktes in *Abschnitt 2.1.2* benötigt wird.

$$\tan \psi = m$$

Eine Gerade  $l$  wird im Polarkoordinatensystem als ein Punkt der Form

$$l = (r, \varphi)$$

beschrieben, wobei  $r$  der senkrechte Abstand der Geraden zum Ursprung  $(0,0)$  und  $\varphi$  der Winkel zwischen x-Achse und der Normalen der Geraden im kartesischen Koordinatensystem ist. Die Ausrichtung der Geraden kann in Polarkoordinaten wie folgt bestimmt werden:

$$\psi = \begin{cases} \varphi + \pi/2, & \text{falls } \varphi \leq 0 \\ \varphi - \pi/2, & \text{sonst} \end{cases}$$

### Liniensegmente

Ein Liniensegment ist eine Gerade, die durch zwei Endpunkte in ihrer Länge beschränkt ist. Wir beschreiben deshalb ein Liniensegment  $s$  durch zwei Endpunkte  $p_1 = (x_1, y_1)$  und  $p_2 = (x_2, y_2)$ :

$$s = (p_1, p_2)$$

Eine Darstellung in Polarkoordinaten wird nicht benötigt.

## 2.2.2 Selbstlokalisierung

Bei der Exploration einer Umgebung soll der Roboter autonom seine Umwelt vermessen. Um die mittels eines Scans erhaltenen Messungen korrekt in eine metrische



Karte einzutragen, benötigt der Roboter zunächst die Kenntnis der eigenen Position in dieser Umgebung. Umgekehrt ist die Existenz einer Karte jedoch Voraussetzung für die Bestimmung der eigenen Position. Dieses Paradoxon wird in der Literatur als „*Simultaneous Localization and Mapping*“ (*SLAM*) bezeichnet und ist ein anspruchsvolles und noch immer nicht vollständig gelöstes Problem in der AMR-Entwicklung. Die zumindest theoretische Lösung ist die schrittweise Erweiterung einer Karte. Es wird zunächst die Roboterposition aufgrund der bisher erstellten Karte bestimmt und anschließend neue Sensorinformationen in das Modell eingetragen. Wir verwenden in dieser Arbeit ein *Rao-Blackwellized Partikel Filter* für die simultane Lokalisation und Kartenerstellung von Hähnel et al. [2003].

Für die korrekte Eintragung der Scanpunkte  $m_i$  eines Scans in ein Modell wird jedem Scan eine Aufnahmeposition  $(x_r, y_r)$  des Roboters und dessen Orientierung  $\theta_r$  zugeordnet. Theoretisch kommt es durch die Eigenbewegung des Roboters während der Aufnahme eines Scans zu einer Verzerrung, wenn nur eine Aufnahmeposition für alle Messwerte bestimmt wird. In der Praxis kann durch die geringe Geschwindigkeit des Roboters und der kurzen Dauer für einen kompletten Scan die Eigenbewegung vernachlässigt werden und wir benötigen nur eine Aufnahmeposition für alle Messwerte eines Scans.

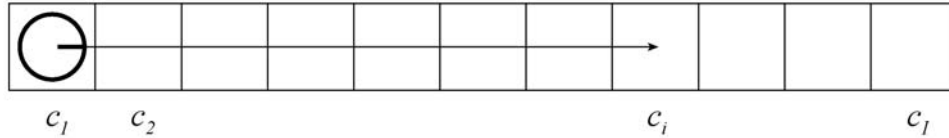
### 2.2.3 Filterung von dynamischen Messpunkten

Im Normalfall soll ein erstelltes Modell bzw. eine erstellte Karte einer Umgebung nur statische Hindernisse beschreiben. Bei der Exploration einer Umgebung werden häufig auch dynamische Hindernisse aufgenommen, wie zum Beispiel Personen, die sich während der Vermessung in der Umgebung bewegen. Werden diese Personen von Roboter gescannt, dann erscheinen sie in der erstellten Karte als Hindernis. Wir versuchen in diesem Abschnitt möglichst alle Messungen von dynamischen Hindernisse zu löschen, damit diese bei der Erstellung des Linienmodells nicht berücksichtigt werden.

Für die Filterung von dynamischen Messpunkten wird eine 2D-Rasterkarte verwendet, die eine Umgebung gitterförmig diskretisiert. Es entsteht eine Umgebungsrepräsentation die aus quadratischen Bereichen besteht. In der Literatur werden solche Karten als „*occupancy grid maps*“ bezeichnet und wurden von Moravec und Elfes [1985] eingeführt. Für unsere Situation erstellen wir ein Gitter, bei dem jede Zelle einen Wert der Wahrscheinlichkeit enthält, den Sensorstrahl an dieser Position zu reflektieren. Die Reflektionswahrscheinlichkeit  $p(r_{x,y})$  jeder Zelle  $(x, y)$  wird anhand der Laserstrahlen der Vermessung berechnet. Mit diesen Wahrscheinlichkeiten können wir die zu entfernenden Messpunkte bestimmen. Unser Verfahren ist mit dem von Thrun et al. [2003] zu vergleichen, wobei wir an einigen Stellen etwas vereinfacht vorgehen.

Typische Diskretisierungen der Umgebung eines Roboters besitzen eine Zellengröße zwischen  $5\text{cm} \times 5\text{cm}$  und  $20\text{cm} \times 20\text{cm}$ . Wir haben für unsere 2D-Rasterkarten eine sehr feine Auflösung von  $5\text{cm} \times 5\text{cm}$  gewählt.

Um die Berechnung der Reflektionswahrscheinlichkeiten zu erläutern, verwenden wir ein vereinfachtes eindimensionales Gittermodell, das in *Abbildung 2.2* dargestellt ist.



**Abbildung 2.2:** Der Roboter befindet sich an Position  $x$  in Zelle  $c_1$  in einer eindimensionalen Umgebung. Der Laserstrahl des Sensors wird in Zelle  $c_i$  reflektiert und überstreicht die Zellen  $c_1$  bis  $c_{i-1}$ .

Wir wollen für jede Zelle  $c_i$ ,  $1 \leq i \leq I$ , die Wahrscheinlichkeit  $p(r_i)$  berechnen, dass der Laserstrahl an dieser Stelle reflektiert wird. Angenommen der Roboter befindet sich an der Position  $x$  und schickt einen Laserstrahl zur Vermessung der Umgebung los. Wir führen zunächst die bedingte Belegungswahrscheinlichkeit  $p(c_i|x,w)$  ein. Der Wert von  $p(c_i|x,w)$  gibt an, mit welcher Wahrscheinlichkeit die Zelle  $c_i$  durch ein Hindernis belegt ist, gegeben der Position  $x$  des Roboters und der Reflektionswahrscheinlichkeiten  $w = \{p(r_i) | 1 \leq i \leq I\}$ . Es sei

$$p(c_i|x,w) = p(r_i) \cdot \prod_{j=1}^{i-1} [1 - p(r_j)], \quad (2.1)$$

wobei  $1 - p(r_j)$  die Wahrscheinlichkeit angibt, dass der Laserstrahl durch Zelle  $c_j$  nicht reflektiert wird. Führen wir  $T$  viele Messungen durch, dann können durch dynamische Objekte und Fehlmessungen Unterschiede in den Messungen auftreten. Wird ein Laserstrahl bei der Messung  $t$ ,  $1 \leq t \leq T$ , durch ein Hindernis in der Zelle  $c_i$  reflektiert, dann bezeichnen wir diese mit  $c_{i,t}$ . Unser Ziel ist es, ein optimales

$$w^* = \arg \max_w \prod_{t=1}^T p(c_{i,t}|x,w) \quad (2.2)$$

zu finden, damit das Produkt der Belegungswahrscheinlichkeiten maximiert wird. Die Gleichung (2.2) lässt sich durch einsetzen von (2.1) wie folgt umformen:

$$\begin{aligned} w^* &= \arg \max_w \prod_{t=1}^T \left[ p(r_{i,t}) \cdot \prod_{j=1}^{(i,t)-1} (1 - p(r_j)) \right] \\ &= \arg \max_w \prod_{t=1}^T [p(r_{i,t})] \cdot \prod_{t=1}^T \prod_{j=1}^{(i,t)-1} [1 - p(r_j)] \end{aligned} \quad (2.3)$$

In (2.3) wird der Term  $1 - p(r_j)$  so oft multipliziert, wie Laserstrahlen bei den

Messungen die Zelle  $c_j$  überstreichen und nicht an dieser Position reflektiert werden. Bezeichnen wir mit  $\alpha_i$  die Anzahl der Laserstrahlen die Zelle  $c_i$  überstreichen und mit  $\beta_i$  die Anzahl der Laserstrahlen die in Zelle  $c_i$  enden, dann kann (2.3) folgendermaßen beschrieben werden:

$$w^* = \arg \max_w \prod_{i=1}^I p(r_i)^{\beta_i} \cdot \prod_{i=1}^I p(1-p(r_i))^{\alpha_i} \quad (2.4)$$

Um uns die Maximierung in (2.4) zu vereinfachen, nehmen wir den Logarithmus dieser Funktion, der aus dem Produkt eine Summe bildet.

$$w^* = \arg \max_w \sum_{i=1}^I (\beta_i \cdot \ln p(r_i)) + \sum_{i=1}^I (\alpha_i \cdot \ln(1-p(r_i)))$$

Das Ergebnis bleibt gleich, da der Logarithmus monoton ist. Für die Berechnung von  $w^* = \{p^*(r_i) | 1 \leq i \leq I\}$  können wir die einzelnen  $p^*(r_i)$  getrennt voneinander bestimmen.

$$p^*(r_i) = \arg \max_{p(r_i)} \beta_i \cdot \ln p(r_i) + \alpha_i \cdot \ln(1-p(r_i)) \quad (2.5)$$

Wir setzen nun die partielle Ableitung nach  $p(r_i)$  des Terms aus (2.5) gleich 0 und erhalten:

$$0 = \frac{\beta_i}{p(r_i)} - \frac{\alpha_i}{(1-p(r_i))} \Leftrightarrow p(r_i) = \frac{\beta_i}{\alpha_i + \beta_i} \quad (2.6)$$

Die Lösung dieser Gleichung ist ein Maximum, da die zweite partielle Ableitung kleiner 0 ist. Für die Berechnung der Reflektionswahrscheinlichkeiten  $p(r_i)$  müssen wir nach (2.6) für jede Zelle  $c_i$  die Anzahl der Fälle bestimmen, in denen der Laserstrahl reflektiert bzw. nicht reflektiert wird.

Kommen wir nun zu unserer eigentlichen 2D-Rasterkarte zurück. Wir müssen für jede Zelle  $(x, y)$  die Reflektionswahrscheinlichkeit  $p(r_{x,y})$  berechnen. Jede Zelle bekommt dafür zwei Zähler  $hit_{x,y}$  und  $miss_{x,y}$ . Um alle Werte der Zähler zu bestimmen, verfolgen wir jeden Laserstrahl ausgehend vom Roboter bis hin zur Reflektion.  $hit_{x,y}$  bzw.  $miss_{x,y}$  soll die Anzahl der Laserstrahlen bestimmen, die in Zelle  $(x, y)$  enden bzw. die Zelle  $(x, y)$  überstreichen. Mit den Werten der Zähler kann nun jeder Zelle  $(x, y)$  deren Reflektionswahrscheinlichkeit

$$p(r_{x,y}) = \frac{hit_{x,y}}{hit_{x,y} + miss_{x,y}} \in [0,1]$$

zugeordnet werden. Für Zellen, bei denen der Nenner gleich 0 ist, können wir keine Reflektionswahrscheinlichkeit angeben. Diese Zellen werden bei der Filterung von dynamischen Messwerten aber auch nicht benötigt, da deren Werte von  $hit_{x,y}$  gleich 0 sind und somit kein einziger Messwert in diesen Zellen liegt.

Um nun zu entscheiden, welche Messwerte gelöscht werden sollen, betrachten wir jede einzelne Zelle einer Rasterkarte. Ist die Reflektionswahrscheinlichkeit  $p(r_{x,y})$  der Zelle  $(x,y)$  kleiner als der Parameter  $DELETE\_GRID\_MIN = 0,5$ , dann haben mehr Laserstrahlen diese Zelle überstrichen, als dass sie von der Zelle reflektiert wurden. Alle Messwerte  $m_i$ , welche in dieser Zelle liegen, können gelöscht werden. Ist  $p(r_{x,y})$  größer gleich  $DELETE\_GRID\_MIN$ , dann bleiben alle Messwerte in dieser Zelle erhalten. Somit können die meisten Messwerte dynamischer Objekte und zum Teil auch Fehlmessungen aus unserer Menge von Messwerten gefiltert werden. Setzen wir den Parameter  $DELETE\_GRID\_MIN$  auf einen kleineren bzw. größeren Wert, dann werden insgesamt mehr bzw. weniger Messwerte gelöscht. Dabei können unter Umständen auch korrekte Messpunkte gelöscht werden. Dies ist der Fall bei Hindernispunkten, deren Zelle von fehlerhaften Laserstrahlen überstrichen werden. Damit nicht zu viele korrekte Messpunkte gefiltert werden, wird in dieser Arbeit ein Wert von 0,3 für  $DELETE\_GRID\_MIN$  gewählt. Bleibt ein dynamisches Objekt länger auf einer festen Position in der Umgebung und der Roboter scannt diese Position nicht zu einem Zeitpunkt, in dem sie frei ist, dann bleiben die Messwerte solch eines dynamischen Objekts erhalten. Das bedeutet für dieses vorgestellte Verfahren, dass nicht alle, aber ein großer Teil von Messwerten dynamischer Hindernisse gelöscht werden können.

## 3 Ein heuristisches Verfahren

In diesem Kapitel wird ein Verfahren zur Generierung von Linienmodellen für die Umgebungen eines Roboters vorgestellt. Bisher haben wir in Kapitel 2 die Umgebung des Roboters exploriert und vermessen, wodurch wir eine Menge von Messpunkten erhalten haben. Diese Menge konnte zu einem Punktemodell der Umgebung zusammengefasst werden. Wir versuchen nun mit diesem heuristischen Verfahren ein Punktemodell in mehreren Schritten in ein Linienmodell zu transferieren, das aus einer Menge von Liniensegmenten besteht. Da ein Linienmodell einem höheren Abstraktionsgrad in der Darstellung der Elemente besitzt als ein Punktemodell, bringt die Transformation einige Probleme mit sich.

Zunächst wird in diesem Kapitel die Geradenextraktion aus einem Punktemodell vorgestellt. Danach wird die Segmentierung einer Geraden und die Berechnung der optimalen Ausrichtung, ein sogenanntes *line-fitting*, beschrieben. Als Letztes versuchen wir die Menge von Liniensegmenten zu verändern.

### 3.1 Geradenextraktion mittels der Hough-Transformation

Die *Hough-Transformation* stellt ein Verfahren zur Detektion kollinearere Punkte dar, welches von Hough [1962] entworfen und später von Duda und Hart [1973] weiterentwickelt wurde. Diese Methode wird vor allem in der Computervision und Bildverarbeitung angewendet, um Strukturen und Muster in Bildern zu erkennen, die in einer geschlossenen, parametrisierbaren Form dargestellt werden können, wie zum Beispiel Geraden und Kreise ([Haberäcker 1995] und [Pitas 2000]). Aber auch in der Robotik ist die Hough-Transformation ein beliebtes Verfahren und wird häufig zum Finden von linearen Hindernissen in Umgebungen verwendet (z.B. [Arras und Siegart 1997], [Pfister et al. 2003] und [Leonard et al. 2001]). Im Folgenden werden wir uns auf das Finden von Geraden konzentrieren.

### 3.1.1 Grundidee der Hough-Transformation

Im kartesischen Koordinatensystem wird eine Gerade häufig durch ihre Steigung  $m$  und den y-Achsenabschnitt  $b$  definiert:

$$y = m \cdot x + b$$

Eine andere Darstellung einer Geraden ist die *Hesse-Normalform*

$$\vec{n} \cdot (\vec{x} - \vec{a}) = 0,$$

wobei  $\vec{a}$  ein Haltepunkt der Geraden ist und  $\vec{n}$  die Normale mit  $|\vec{n}| = 1$ . Aus  $|\vec{n}| = 1$  folgt

$$\vec{n} = \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix}.$$

Dabei ist  $\varphi$  der Winkel zwischen der positiven x-Achse und der Normalen  $\vec{n}$ . Es gilt dann:

$$\vec{n} \cdot (\vec{x} - \vec{a}) = \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix} \cdot \left( \begin{pmatrix} x \\ y \end{pmatrix} - r \cdot \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix} \right)$$

Der senkrechte Abstand zwischen dem Ursprung  $(0,0)$  und der Geraden ist  $r$ . Eine Gerade wird also durch  $(r, \varphi)$  beschrieben.

Für alle Punkte  $(x, y)$  auf einer Geraden  $l_0$  gilt nun:

$$r_0 = x \cos \varphi_0 + y \sin \varphi_0$$

Somit kann eine Gerade in der xy-Ebene auf einem Punkt in der  $r\varphi$ -Ebene abgebildet werden, wobei  $\varphi \in [0, \pi)$  und  $r \in \mathbb{R}$ . Auf diese Weise lassen sich Daten zu einer einfacheren Analyse aufbereiten. Auf der Grundlage dieser Gerade-Punkt-Transformation lässt sich die Hough-Transformation wie folgt mathematisch definieren:

**Definition:**

Die Hough-Transformation bildet einen Punkt  $(x, y) \in \mathbb{R}^2$  in den Funktionsraum der sinusförmigen Funktion ab:

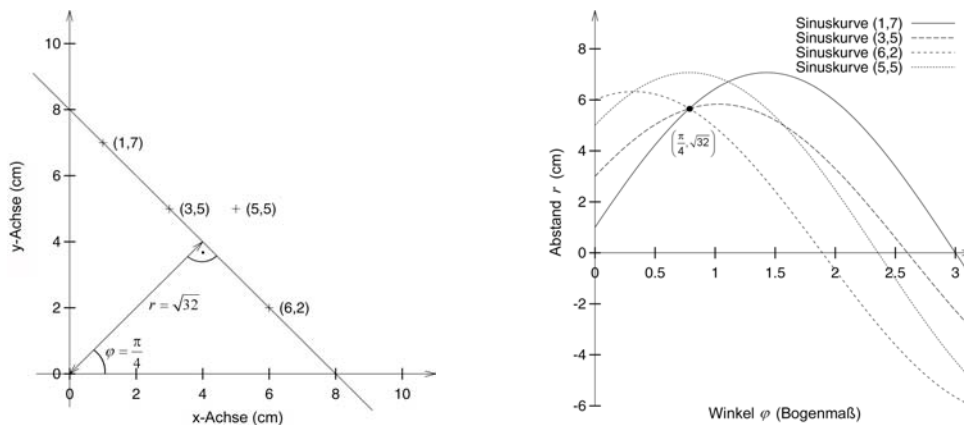
$$h : (x, y) \rightarrow r = x \cos \varphi + y \sin \varphi$$

wobei  $\varphi \in [0, \pi)$  und  $r \in [-r_{\max}, r_{\max}]$ .

Dabei ist  $r_{\max}$  der maximale senkrechte Abstand einer Geraden zum Ursprung des kartesischen Koordinatensystems. Die  $r\varphi$ -Ebene wird auch als Hough-Raum bezeichnet.

Betrachten wir einen beliebigen Punkt  $(x, y)$  in der  $xy$ -Ebene. Durch diesen einzelnen Punkt können unendlich viele verschiedene Geraden verlaufen. Mittels der Hough-Transformation können alle Geraden, die durch den Punkt  $(x, y)$  verlaufen, in einer sinusförmigen Kurve  $r = x \cos \varphi + y \sin \varphi$  in der  $r\varphi$ -Ebene dargestellt werden.

Seien nun  $(x_i, y_i)$  mehrere Punkte auf einer Geraden  $l_0$ . Zu jedem Punkt gehört eine Sinuskurve in der  $r\varphi$ -Ebene. Diese Kurven haben alle genau einen gemeinsamen Schnittpunkt  $(r_0, \varphi_0)$ . Die Sinuskurven aller weiteren Punkte  $(x, y)$  auf der Geraden  $l_0$  müssen sich in dem Schnittpunkt  $(r_0, \varphi_0)$  schneiden. Liegt ein Punkt nicht auf der Geraden  $l_0$ , dann verläuft seine Sinuskurve nicht durch den Schnittpunkt  $(r_0, \varphi_0)$  der anderen Kurven. Somit repräsentiert der Schnittpunkt  $(r_0, \varphi_0)$  der Sinuskurven  $r = x_i \cos \varphi + y_i \sin \varphi$  die Gerade  $l_0$ . *Abbildung 3.1* veranschaulicht die beschriebenen Grundlagen der Hough-Transformation.



**Abbildung 3.1:** Das linke Schaubild zeigt vier Punkte, von denen drei auf und einer nicht auf einer Geraden in der  $xy$ -Ebene liegen. Im rechten Schaubild werden die Sinuskurven der vier Punkte in der  $r\varphi$ -Ebene dargestellt. Der gemeinsame Schnittpunkt der drei Kurven stellt die Gerade im linken Bild dar.

Es ist somit möglich, die Anzahl der auf einer Geraden liegenden Punkte zu bestimmen. Die Anzahl der Punkte auf einer Geraden  $l_0$  entspricht der Anzahl der Sinuskurven, die sich im Punkt  $(r_0, \varphi_0)$  in der  $r\varphi$ -Ebene schneiden.

### 3.1.2 Der Hough-Algorithmus

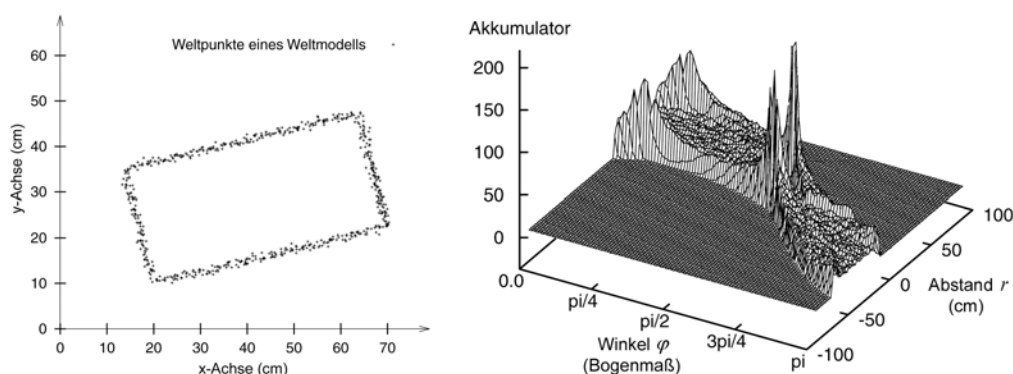
Da es nicht möglich ist, die Anzahl der Schnittpunkte der Sinuskurven in jedem beliebigen Punkt der  $r\varphi$ -Ebene zu zählen, muss die  $r\varphi$ -Ebene für  $-r_{max} \leq r \leq r_{max}$  und  $0 \leq \theta \leq \pi$  diskretisiert werden. Der diskretisierte Hough-Raum wird auch Akkumulator genannt und wird mittels einer Akkumulatormatrix  $a(i, j)$  dargestellt.

### 3 Ein heuristisches Verfahren

Soll der Hough-Raum durch eine  $(R+1) \times (C+1)$  Akkumulatormatrix diskretisiert werden, so stellt jede Spalte  $j$ ,  $0 \leq j \leq C$ , einen Winkel  $j \frac{\pi}{C}$  dar. Jede Reihe  $i$ ,  $0 \leq i \leq R$ , repräsentiert einen Abstand  $(i \cdot \frac{2r_{max}}{R} - r_{max})$ . Eine Akkumulatorzelle  $a(i, j)$  dient als Zähler für den Punkt  $(i \cdot \frac{2r_{max}}{R} - r_{max}, j \frac{\pi}{C})$  in der  $r\varphi$ -Ebene. Je größer die Anzahl der Zeilen und Spalten gewählt wird, um so mehr verschiedene Punkte im Hough-Raum können dargestellt werden. Jede Zelle stellt genau eine Gerade in der  $xy$ -Ebene dar. In unseren Versuchen haben wir 181 Spalten und  $2 \cdot r_{max} + 1$  Zeilen verwendet, womit wir eine Genauigkeit von  $1^\circ$  in den Winkeln  $\varphi$  und eine Genauigkeit von 1 cm in den Abständen  $r$  der Geraden erhalten.

Zu Beginn der Hough-Transformation wird jede Zelle der Akkumulatormatrix mit dem Wert 0 initialisiert. Im eigentlichen Transformationsschritt wird für jeden Messpunkt  $m_i = (x_i, y_i)$  einer Umgebung der gesamte diskretisierte Wertebereich von  $\varphi$  durchlaufen. Für jedes  $\varphi$  wird der zugehörige Wert  $r$  mittels der Sinuskurve  $r = x_i \cos \varphi + y_i \sin \varphi$  berechnet. Dabei wird jeweils die Akkumulatorzelle um den Wert 1 erhöht, welche die berechnete Gerade  $(r, \varphi)$  repräsentiert. Hierbei ist zu beachten, dass der berechnete Wert  $r$  auf einen diskretisierten Wert gerundet werden muss, damit eine Erhöhung eines Wertes in die Matrix möglich ist. Die entstandene Akkumulatormatrix entspricht einem Histogramm. Die Transformation der Messpunkte in den Hough-Raum ist somit abgeschlossen.

Nach Anwendung dieses Verfahrens stellt jeder Wert der Akkumulatormatrix die Anzahl der Messpunkte dar, welche auf einer Geraden  $(r, \varphi)$  liegen. Ein höher Eintrag im Akkumulator deutet auf eine große Menge von kollineareren Punkte hin und lässt uns auf eine lineare Struktur in einer Umgebung schließen. Ein simuliertes Beispiel in *Abbildung 3.2* verdeutlicht diesen Zusammenhang.



**Abbildung 3.2:** Die linke Abbildung zeigt ein künstlich erstelltes Punktemodell mit vier linearen Strukturen. Der dazugehörige Akkumulatorraum ist rechts dargestellt. Die vier verschiedenen lokalen Maxima weisen auf die vier linearen Strukturen im Punktemodell hin.



### 3.1.3 Erweiterung durch die lokale Ausrichtung

Bei der Definition der Hough-Transformation wird ein Punkt in eine sinusförmige Funktion abgebildet, wobei der Wertebereich des Winkels  $\varphi$  durch  $\varphi \in [0, \pi)$  beschränkt ist. Bei der Erstellung der Akkumulatormatrix muss nach dieser Definition der komplette diskretisierte Wertebereich von  $\varphi$  für jeden Messpunkt berücksichtigt werden. Mit einer Auflösung von  $1^\circ$  bei der Diskretisierung müssen wir für einen Messpunkt jeweils 181 verschiedene Zellen in der Akkumulatormatrix berechnen und erhöhen. Berücksichtigen wir die lokale Ausrichtung eines Messpunktes, dann können wir diese Anzahl von Berechnungen pro Messpunkt verringern. Um dies zu realisieren, verändern wir den Wertebereich des Winkels  $\varphi$  bei der Definition der Hough-Transformation. Sei  $\omega'$  die lokale Ausrichtung eines Messpunktes in der  $r\varphi$ -Ebene, dann beschränken wir die Transformation eines Punktes auf einen Wertebereich

$$\varphi \in \begin{cases} \left[ \omega' - \frac{\pi}{k}, \omega' + \frac{\pi}{k} \right] & , \text{ falls } \frac{\pi}{k} \leq \omega' < \frac{k-1}{k} \pi \\ \left[ 0, \omega' + \frac{\pi}{k} \right] \cup \left[ \frac{k-1}{k} \pi + \omega', \pi \right) & , \text{ falls } 0 \leq \omega' < \frac{\pi}{k} \\ \left[ \omega' - \frac{\pi}{k}, \pi \right) \cup \left[ 0, \omega' - \frac{k-1}{k} \pi \right] & , \text{ falls } \frac{k-1}{k} \pi \leq \omega' < \pi \end{cases}$$

wobei  $k > 2$  ist. Mit dieser Einschränkung von  $\varphi$  werden bei der Transformation nur solche Zellen der Akkumulatormatrix erhöht, deren Ausrichtung der Gerade um weniger als  $\pi/k$  von der lokalen Ausrichtung des Punktes abweichen. Die lokale Ausrichtung beschreibt ungefähr die Lage der linearen Struktur eines Messpunktes, und deshalb macht diese Einschränkung des Wertebereiches Sinn, da der Akkumulator die lineare Struktur durch ein lokales Maximum darstellen soll. Durch diesen Ansatz können wir die Zellen der Akkumulatormatrix gezielter vergrößern. Der Wert eines lokalen Maximums wird dadurch zwar nicht vergrößert, aber dafür wird das gesamte Niveau der Matrix verkleinert und ein lokales Maximum kann besser erkannt werden. Da die lokale Ausrichtung  $\omega'$  nur ungefähr den Winkel  $\varphi$  der linearen Struktur in der  $r\varphi$ -Ebene beschreibt, darf der Wert für  $k$  nicht zu groß gewählt werden, damit nicht zu wenig Zellen im Akkumulator berücksichtigt werden. Benützen wir zum Beispiel  $k=9$ , dann werden für einen Messpunkt alle Geraden mit einer maximalen Abweichung von  $\pi/9 \equiv 20^\circ$  bei der Transformation berücksichtigt. Damit müssen wir bei einer Auflösung von  $1^\circ$  nur noch 40 verschiedene Akkumulatorzellen pro Messpunkt berechnen und erhöhen. Im Allgemeinen ergibt dies eine Beschleunigung bei der Berechnung der kompletten Akkumulatormatrix um  $k/2$ . Ob wir in der Praxis tatsächlich mit dieser Erweiterung unsere Linienmodelle verbessern können, wird in *Abschnitt 5.2.1* analysiert.

#### 3.1.4 Extraktion von Geraden

Nach Durchführung der Hough-Transformation für alle Messpunkte eines Punktemodells wird nun der Akkumulator analysiert. Jede Akkumulatorzelle mit  $a(i, j) > 1$  stellt eine Gerade dar, auf der mindestens zwei Messpunkte liegen. Das Maximum der Matrix sollte die signifikanteste lineare Struktur in einer Umgebung repräsentieren. Diese Struktur zeichnet sich durch die größte Menge kollinearere Messpunkte aus.

Um nun mehrere Geraden aus dem erstellten Akkumulator zu extrahieren, wenden wir ein iteratives Verfahren an. Als erstes wird das Maximum der Matrix bestimmt und die Gerade berechnet. Anhand dieser Geraden wird ein Liniensegment erstellt, das eine lineare Struktur der Umgebung beschreiben soll. Diesem Liniensegment wird eine Menge von Messpunkten zugeordnet. Die Berechnung der Liniensegmente, die Zuordnung der Punkte und die dabei entstehenden Probleme werden in den folgenden Abschnitten genauer beschrieben. Nachdem wir nun wissen, welche Punkte dem erstellten Liniensegment zugeordnet wurden, können alle Sinuskurven der zugeordneten Punkte von der aktuellen Akkumulatormatrix entfernt werden, damit sie im nächsten Schritt das Ergebnis nicht beeinflussen. Dies geschieht analog zur eigentlichen Hough-Transformation, nur diesmal werden die Zellen der Matrix nicht um der Wert 1 erhöht, sondern erniedrigt. Dabei wird nicht nur das Maximum der Matrix entfernt, sondern auch das Gesamtniveau des Histogramms wird gesenkt. Dadurch muss die Matrix im nächsten Iterationsschritt nicht nochmals erneut aus allen verbleibenden Punkten berechnet werden. Dieser Prozess wird solange iteriert, bis entweder alle Messpunkte einem Liniensegment zugeordnet wurden oder ein anderes Abbruchkriterium erfüllt ist (vgl. *Abschnitt 3.4*).

Die Laufzeit der Extraktion von Geraden mittels der Hough-Transformation ist durch zwei Komponenten geprägt; zum einen von der Berechnung der Akkumulatormatrix und zum anderen von der Bestimmung des Maximums der Matrix. Sei  $n$  die Anzahl der Messpunkte und  $c$  die Anzahl der Spalten der Matrix, dann wird für die Berechnung der Matrix eine Zeit von  $O(c \cdot n)$  benötigt, da für die Transformation eines Messpunktes für jede Spalte eine Zellen berechnet werden muss. Jede Sinuskurve eines Punktes wird genau zweimal berechnet und in die Akkumulatormatrix übertragen. Das erste Mal muss die Sinuskurve bei der Erstellung des Akkumulators berechnet werden. Wenn der Messpunkt einem Liniensegment zugeordnet wird, muss die Kurve ein zweites Mal berechnet und vom Akkumulator abgezogen werden. Die Laufzeit für die Bestimmung des Maximums ist von der Größe der Matrix abhängig. Die Genauigkeit der Diskretisierung und vor allem die Größe einer vermessenen Umgebung bestimmen das Ausmaß der Matrix. Für eine  $c \times r$  Matrix ergibt sich eine Laufzeit von  $O(c \cdot r)$  für die Bestimmung eines Maximums.

## 3.2 Übergang von einer Geraden zu einem Liniensegment

Die gefundenen Geraden der Hough-Transformation sind unendlich lang. Da wir an Segmenten für unser Linienmodell interessiert sind, muss eine Gerade mittels zweier Endpunkte in ihrer Länge beschränkt werden. Dafür werden zuerst Messpunkte einer Geraden zugeordnet. Anhand dieser Punkte kann dann eine Gerade in ein oder mehrere Segmente unterteilt werden. Da wir aus jeder Geraden nur ein Liniensegment erhalten wollen, filtern wir das signifikanteste Liniensegment heraus.

### 3.2.1 Zuordnung von Punkten zu einer Geraden

Eine extrahierte Gerade aus der Akkumulatormatrix repräsentiert eine lineare Struktur, wie zum Beispiel eine gerade Wand. In jedem Iterationsschritt werden einer extrahierten Geraden aus dem Hough-Raum anschließend Messpunkte zugeordnet. Diese Messpunkte sollten im idealen Fall durch die gleiche lineare Struktur bei der Vermessung der Umgebung entstanden sein.

Für die Extraktion einer Geraden  $l_0$  aus dem Hough-Raum sind alle kollinearen Punkte verantwortlich, die auf dieser Geraden liegen. Nur diese Punkte tragen dazu bei, dass sich an dieser Stelle in der Akkumulatormatrix ein Maximum bildet. Durch eine gewisse Fehlerrate in der Lage der Messpunkte liegen nicht alle Punkte einer linearen Struktur genau auf der Geraden  $l_0$ , sondern weichen etwas ab. Deshalb werden in unserem Verfahren nicht nur alle Punkte die auf  $l_0$  liegen dieser Geraden zugeordnet, sondern alle die Punkte, deren Abstand kleiner ist als der Schwellenwert  $ASSIGN\_DIST\_MAX$ <sup>4</sup>. Die Wahl des Wertes dieses Parameters ist maßgeblich für die Güte des am Ende erstellten Linienmodells verantwortlich. Wird  $ASSIGN\_DIST\_MAX$  zu groß gewählt, dann können Messpunkte der Geraden  $l_0$  zugeordnet werden, die nicht zur linearen Struktur dazugehören, die  $l_0$  repräsentiert. Wird hingegen  $ASSIGN\_DIST\_MAX$  zu klein gewählt, dann können zu wenig Messpunkte der Geraden  $l_0$  zugeordnet werden. Die nicht zugeordneten Punkte bleiben im nächsten Iterationsschritt im Akkumulator und können eine neue Gerade erzeugen, welche die gleiche lineare Struktur darstellt wie  $l_0$ . In *Abschnitt 5.2.1* untersuchen wir den Einfluss dieses Parameters auf die erstellten Modelle und versuchen für die Praxis geeignete Werte zu finden.

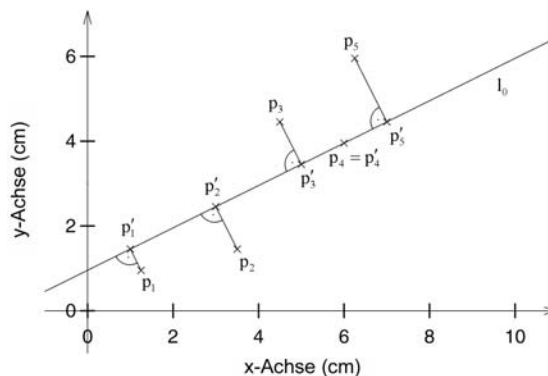
---

<sup>4</sup> Alle in diesen Kapitel benötigten Parameter sind im Anhang zusammenfassend aufgelistet und werden dort kurz beschrieben (Seite 112).

### 3.2.2 Segmentierung einer Geraden

Nachdem wir einer Geraden  $l_0$  eine Punktmenge  $P = \{p_1, \dots, p_n\}$ ,  $n \geq 2$ , zugeordnet haben, kann diese anhand der Lage der Punkte in Segmente aufgeteilt werden. Bei der Segmentierung können aus einer Geraden ein oder mehrere Liniensegmente  $s_j$  entstehen, die jeweils eine lineare Struktur beschreiben sollen. Ein Liniensegment  $s_j$  wird durch zwei Endpunkte  $p_{j,1}$  und  $p_{j,2}$  beschrieben.

Zunächst werden alle zugeordneten Punkte aus  $P$ , wie in *Abbildung 3.3*, auf die Gerade  $l_0$  abgebildet. Dadurch entsteht eine neue Punktmenge  $P' = \{p'_1, \dots, p'_n\}$ .



**Abbildung 3.3:** Die Punktmenge  $P = \{p_1, \dots, p_5\}$  wird auf die Gerade  $l_0$  abgebildet und es entsteht die Punktmenge  $P' = \{p'_1, \dots, p'_5\}$ .

Als Nächstes sortieren wir die neu erstellte Punktmenge  $P'$  nach der Lage der Punkte. Ist die Steigung  $m$  der Geraden  $l_0 : y = m \cdot x + b$  eher flach, also  $|m| < 1$ , dann wird die Punktmenge  $P'$  nach ihren x-Koordinaten sortiert. Ist die Steigung eher steil, also  $|m| \geq 1$ , dann wird nach den y-Koordinaten sortiert. Durch diese Unterscheidung werden Fehler bei senkrechten und waagerechten Geraden vermieden. Die ermittelte sortierte Punktmenge bezeichnen wir als  $\tilde{P} = \{\tilde{p}_1, \dots, \tilde{p}_n\}$ . Im Folgenden nehmen wir an, dass die Gerade  $l_0$  eine eher flache Steigung besitzt und somit gilt für die x-Werte der Punktmenge  $\tilde{P}$ :  $x_1 \leq x_2 \leq \dots \leq x_n$ . Mit dieser Punktmenge können wir jetzt mit der eigentlichen Segmentierung beginnen, die durch den *Algorithmus 2* beschrieben wird.

Für die Segmentierung einer Geraden wird der *segment*-Funktion die komplette Punktmenge  $\tilde{P}$  übergeben. Um nun zu entscheiden, ob zwei Punkte durch die gleiche lineare Struktur in der Umgebung entstanden sind, führen wir den Parameter *SEG\_DIST\_MAX* ein. Der Wert *SEG\_DIST\_MAX* ist der maximale Abstand den zwei aufeinander folgender Punkte haben dürfen, damit sie zur gleichen linearen Struktur zugeordnet werden. Ist also der Abstand  $abstand(\tilde{p}_i, \tilde{p}_{i+1})$  der Punkte  $\tilde{p}_i$  und  $\tilde{p}_{i+1}$  kleiner gleich *SEG\_DIST\_MAX*, dann gehören beide Punkte zur gleichen linearen Struktur. Mit der for-Schleife werden nun nacheinander so lange alle Abstände berechnet, bis der

**Algorithmus 2:**  $segment(\tilde{P})$ **Eingabe:** Menge  $\tilde{P} = \{\tilde{p}_1, \dots, \tilde{p}_n\}$  von sortierten Punkten**Ausgabe:** Menge  $S$  von Liniensegmenten**Ablauf:**  $S = \emptyset$  $start = 1$ **for**  $i = 1$  **to**  $n - 1$  **do**    **if**  $abstand(\tilde{p}_i, \tilde{p}_{i+1}) > SEG\_DIST\_MAX$  **then**        **if**  $start \neq i$  **then**             $S = S \cup s(\tilde{p}_{start}, \tilde{p}_i)$         **endif**         $start = i + 1$     **endif****endfor****if**  $start \neq n$  **then**     $S = S \cup s(\tilde{p}_{start}, \tilde{p}_n)$ **endif****return**  $S$ 

Abstand zweier Punkte  $\tilde{p}_i$  und  $\tilde{p}_{i+1}$  größer  $SEG\_DIST\_MAX$  ist. Diese Punkte gehören somit zu zwei verschiedenen linearen Strukturen. Die Punkte  $\tilde{p}_{start}$  bis  $\tilde{p}_i$  gehören hingegen zur gleichen linearen Struktur und bilden ein Liniensegment  $s$ , das durch die Endpunkte  $(\tilde{p}_{start}, \tilde{p}_i)$  beschrieben wird. Bevor ein Liniensegment entsteht wird geprüft, ob  $\tilde{p}_{start}$  und  $\tilde{p}_i$  nicht der gleiche Punkt ist, denn ein Punkt allein kann kein Liniensegment bilden. Diese Einteilung in Liniensegmente stellt die Segmentierung der Geraden  $l_0$  dar und wird solange durchgeführt, bis der letzte Punkt  $\tilde{p}_n$  berücksichtigt wurde. Am Anfang der Segmentierung sind alle Punkte  $p_i$ ,  $1 \leq i \leq n$ , der Geraden  $l_0$  zugeordnet. Für den späteren Verlauf unseres Verfahrens ist es wichtig, dass den Liniensegmenten die Punkte  $p_i$  zugeordnet werden, die für ihre Entstehung verantwortlich sind.

Diese Zuordnung kann bei der Berechnung der Segmente im *Algorithmus 2* durchgeführt werden. Nach der Segmentierung sind die einzelnen Punkte  $p_i$  jeweils genau einem Liniensegment zugeordnet.

Der Wert des Parameters  $SEG\_DIST\_MAX$  ist von der Dichte der Lage der Messpunkte im Punktemodell abhängig. Werden bei der Exploration einer Umgebung die Scans schnell hintereinander durchgeführt, so liegen die einzelnen Messpunkte der linearen Strukturen dicht beieinander. Dies ist natürlich auch von der Geschwindigkeit des Roboters abhängig. Eine größere Anzahl von Punkten in einem Punktemodell lässt uns die linearen Strukturen besser finden, da die Abstände zwischen den einzelnen

sortierten Punkten bei der Segmentierung geringere sind. Für die Praxis geeignete Werte für  $SEG\_DIST\_MAX$  und der Einfluss dieses Parameters auf die Linienmodelle wird im *Abschnitt 5.2.1* behandelt.

#### 3.2.3 Filterung der Liniensegmente

Direkt nach jeder Extraktion einer Geraden  $l_0$  aus dem Akkumulatorraum wird deren Segmentierung durchgeführt. Bei jeder Extraktion wollen wir eigentlich nur jeweils die zu diesem Zeitpunkt der Iteration signifikanteste lineare Struktur durch ein Liniensegment beschreiben. Da einer Geraden nicht nur die Messpunkte der signifikanten linearen Struktur zugeordnet werden, sondern alle Punkte im Punktemodell die nah genug an der Geraden liegen, können bei der Segmentierung mehrere Liniensegmente entstehen. Aus dieser Menge müssen wir nun ein Liniensegment  $s_0$  herausfiltern, das die signifikanteste lineare Struktur repräsentiert.

Um  $s_0$  zu bekommen, wird das Liniensegment mit den meisten zugeordneten Punkte gesucht, welches bei der Segmentierung der Linie  $l_0$  entstanden ist. Die Punktmenge von  $s_0$  hat mehr als alle anderen einzelnen Punktemengen dazu beigetragen, dass die Linie  $l_0$  aus dem Akkumulator extrahiert wurde.

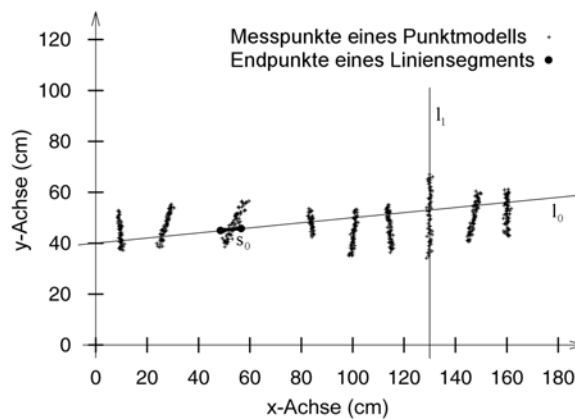
Damit jeder Punkt nur genau einem Liniensegment zugeordnet wird, muss die Punktmenge von  $s_0$  anschließend aus dem Punktemodell genommen werden. Des Weiteren müssen diese Punkte, wie in *Abschnitt 3.1* beschrieben, aus dem Akkumulator heraus transformiert werden. Alle anderen Punktemengen, die den anderen Liniensegmenten der Geraden  $l_0$  zugeordnet wurden und somit nicht zu der signifikanten linearen Struktur gehören, bleiben im Akkumulator. Sie werden bei der nächsten Extraktion einer Geraden aus dem Akkumulator erneut berücksichtigt.

Im Verlauf des ganzen Verfahrens der Extraktion von Liniensegmenten kann es zu verschiedenen Situationen kommen, bei denen die bisherige Segmentierung und anschließende Filterung nicht korrekt abläuft. Ein Beispiel dafür ist in *Abbildung 3.4* dargestellt. Dies passiert vor allem bei sehr großen Punktemodellen, wenn gegen Ende des ganzen Verfahrens nur noch Punkte von vielen kleinen linearen Strukturen übrig geblieben sind. Im Beispiel aus *Abbildung 3.4* wird die Gerade  $l_0$  aus dem Akkumulator extrahiert, da auf dieser Geraden die meisten Punkte liegen. Die Gerade  $l_1$  beschreibt eigentlich die signifikanteste lineare Struktur in dieser Welt, aber auf ihr liegen insgesamt zu wenig kollineare Punkte, damit an dieser Stelle ein Maximum im Akkumulator entsteht. In dieser Situation entsteht ein fehlerhaftes Liniensegment  $s_0$ . Um dies zu vermeiden, wird ein weiterer Parameter  $SEG\_LENGTH\_MIN$  eingeführt, der die Mindestlänge eines Liniensegments angibt. Wird also ein Liniensegment  $s_0$  extrahiert dessen Länge kleiner als  $SEG\_LENGTH\_MIN$  ist, dann verwerfen wir dieses

Segment, und alle diesem  $s_0$  zugeordneten Punkte dürfen nicht aus dem Akkumulator transformiert werden. Da in diesem Iterationsschritt der Akkumulator unverändert geblieben ist, würde im nächsten Schritt erneut die Gerade  $l_0$  extrahiert werden und bei der Segmentierung das gleiche  $s_0$  entstehen wie zuvor. Daher setzen wir den Wert der Geraden  $l_0$  im Akkumulator auf 0, damit diese Gerade nicht wiederholt extrahiert wird. Die Einführung von  $SEG\_LENGTH\_MIN$  hat jedoch zur Folge, dass keine Liniensegmente kleiner als der Wert von  $SEG\_LENGTH\_MIN$  entstehen können. Der Wert dieses Parameters sollte nicht zu groß gewählt werden, damit wir auch kleine lineare Strukturen durch Liniensegmente beschreiben können.

Mit einer geeigneten Wahl des Wertes für den Parameter  $SEG\_LENGTH\_MIN$  können die meisten fehlerhaften Liniensegmente herausgefiltert werden. Die Bestimmung eines Wertes für diesen Parameter und dessen Einfluss auf ein erstelltes Linienmodell wird im späteren *Abschnitt 5.2.1* erläutert.

Alle linearen Strukturen mit einer kleineren Länge als  $SEG\_LENGTH\_MIN$  können dann mit der Hough-Transformation allein nicht bestimmt werden. Um diese Strukturen dennoch durch ein Liniensegment zu beschreiben, haben wir den bisherigen Algorithmus um ein in *Abschnitt 3.5.3* beschriebenes Verfahren erweitert.



**Abbildung 3.4:** Ein simuliertes Punktemodell mit einigen kurzen linearen Strukturen; Die Gerade  $l_0$  wird aus dem Akkumulator extrahiert und ein fehlerhaftes Liniensegment  $s_0$  entsteht.

### 3.3 Anpassung eines Liniensegments an eine Punktemenge

Mittels der Extraktion einer Geraden  $l_0$  aus dem Hough-Raum und deren Segmentierung bekommen wir ein Liniensegment  $s_0$  mit zugeordneten Punkten, das eine lineare Struktur der Umgebung in unserem Linienmodell darstellen soll. Die Lage des Liniensegments ist zu diesem Zeitpunkt des Verfahrens noch nicht optimal zur linearen Struktur ausgerichtet. Ein Grund dafür ist der diskretisierte Akkumulator. Ein weitaus gravierenderer Grund sind die Messpunkte die auf der Geraden  $l_0$  liegen und nicht dem Liniensegment  $s_0$  zugeordnet wurden. Diese Punkte haben mit dazu beigetragen, dass die Gerade  $l_0$  extrahiert wurde. Sie sind aber auch für die momentane Ausrichtung des Liniensegments  $s_0$  mit verantwortlich, obwohl sie zu ganz anderen linearen Strukturen gehören als  $s_0$  repräsentiert. Deshalb passen wir ein Liniensegment nach der Segmentierung an die zugeordnete Punktemenge an und optimieren somit die Lage des Liniensegments. In diesem Schritt entsteht das Liniensegment  $s_0^*$ , das eine lineare Struktur noch besser repräsentiert als  $s_0$  zuvor. Dabei ist zu beachten, dass bei dieser Anpassung eine Gerade und nicht ein Liniensegment verwendet wird. Solch eine Anpassung einer Geraden an eine Menge von Punkten wird im Allgemeinen als *line-fitting* bezeichnet. Anschließend müssen wir die angepasste Gerade wieder segmentieren, damit wir das Liniensegment  $s_0^*$  bekommen.

Im Folgenden werden zwei verschiedene Verfahren zur Anpassung einer Geraden an eine Punktemenge vorgestellt und deren Vor- und Nachteile erläutert.

#### 3.3.1 Line-fitting als lineare Regression

Bei diesem Ansatz wird davon ausgegangen, dass wir eine Menge von Messpunkten  $P = \{(x_1, y_1), \dots, (x_n, y_n)\}$  besitzen, an die eine Gerade  $y = m \cdot x + b$ , auch Regressionsgerade genannt, bestmöglich angepasst werden soll. Es wird angenommen, dass in Abhängigkeit eines gegebenen Wertes  $x_i$  ein  $y_i$  gemessen wird und somit die Menge  $P$  entsteht. Wenn nun die Punkte aus  $P$  eine Gerade relativ gut beschreiben, dann könnte man einen linearen Zusammenhang zwischen den Werte  $x_i$  und  $y_i$  vermuten. Mathematisch bedeutet dies, dass die lineare Beziehung der einzelnen Wertepaare  $(x_i, y_i)$  durch den Messfehler  $f_i$  gestört ist und es gilt:

$$y_i = m \cdot x_i + b + f_i, \quad \forall i$$

Der ideale Wert von  $y_i$  bei gegebenem  $x_i$  wäre  $y_i = m \cdot x_i + b$ . Um nun die Parameter der Geraden zu finden, die am besten zu allen  $y_i$  passen, versucht man die Summe  $Q$  der Quadrate der Messfehler zu minimieren. Die Summe  $Q$  lässt sich wie folgt berechnen:



$$Q = \sum_{i=1}^n f_i^2 = \sum_{i=1}^n (y_i - m \cdot x_i - b)^2 \quad (3.1)$$

Um zur ursprünglichen Situation in unserem Linienmodell zurückzukommen, beschreibt (3.1) die Summe der quadratischen y-Abstände der Messungen zur Geraden. Diese Fehlerfunktion wird beim *line-fitting* in der Literatur in vielen Fällen (z.B. [Press et al. 1992] und [Krengel 1991]) verwendet und nicht die Summe der quadratischen senkrechten Abstände der Messungen zur Geraden berücksichtigt (vgl. *Abschnitt 3.3.2*). Das Ergebnis ist in der Praxis trotzdem akzeptabel (z.B. [Gonzalez et al. 1994] und [Gutmann et al. 2002]) und wird deshalb an dieser Stelle beschrieben.

Für die Bestimmung der Parameter  $(m, b)$  der Geraden muss (3.1) minimiert werden. Dafür setzen wir die partiellen Ableitungen von  $Q$  nach  $m$  und  $b$  gleich 0 und erhalten folgendes Gleichungssystem:

$$\begin{aligned} -2 \sum_{i=1}^n (y_i - m \cdot x_i - b) &= 0 \\ -2 \sum_{i=1}^n x_i (y_i - m \cdot x_i - b) &= 0 \end{aligned}$$

Durch Umformen lassen sich die Werte für die beiden Unbekannten  $m$  und  $b$  berechnen durch:

$$\begin{aligned} b &= \frac{\left( n \cdot \sum_{i=1}^n x_i \cdot y_i \right) - \left( \sum_{i=1}^n x_i \cdot \sum_{i=1}^n y_i \right)}{\sum_{i=1}^n x_i^2 - \left( \sum_{i=1}^n x_i \right)^2} \\ m &= \sum_{i=1}^n y_i - b \cdot \sum_{i=1}^n x_i \end{aligned}$$

Ist die angepasste Gerade in der Praxis eine Senkrechte, dann ist bei der Berechnung von  $b$  der Nenner gleich 0. Um dies zu vermeiden, versuchen wir das Ergebnis im Voraus abzuschätzen; dies können wir dadurch erreichen, indem wir die Steigung der Geraden  $l_0$  vor der Anpassung betrachten. Ist die Steigung  $|m| \leq 1$ , dann wird das Ergebnis eine eher flach verlaufende Gerade und wir können die Anpassung wie beschrieben durchführen. Ist die Steigung  $|m| > 1$ , dann drehen wir das Koordinatensystem um  $90^\circ$  und berechnen die neuen Werte für  $m$  und  $b$ . Ist die Steigung  $m = 0$ , dann entsteht bei der anschließend Transformation in das alte Koordinatensystem eine senkrechte Gerade, welche bei der anschließenden Segmentierung als Sonderfall betrachtet wird.

### 3.3.2 Line-fitting als nicht-lineare Regression

Bei der Vermessung der Umgebung entstehen Messwerte mit einer gewissen Unsicherheit in den x/y-Werten. Diese Unsicherheit wird unter anderem durch den Lasersonar und die Selbstlokalisierung verursacht. Es stellt sich daher die Frage, welche Fehlerfunktion bei der Regression minimiert werden soll, um eine Gerade an eine Menge von Punkten anzupassen. Wir könnten versuchen, eine algebraisch korrekte Fehlerfunktion herzuleiten, aber da unser eigentliches Problem eher von geometrischem Charakter ist, bilden wir daher eine geometrische Fehlerfunktion. In der Robotik wird das folgende *line-fitting*-Verfahren häufig eingesetzt, so zum Beispiel auch von Arras und Siegwart [1997], Lu und Milios [1994] und Pfister et al. [2003].

Ist eine Menge von Messpunkten  $P = \{p_1, \dots, p_n\}$  und eine Gerade  $l_0 : y = m \cdot x + b$  gegeben, dann benützen wir die Fehlerfunktion

$$\chi^2(m, b) = \sum_{i=1}^N d(p_i, l_0)^2, \quad (3.2)$$

wobei  $d(p_i, l_0)$  der senkrechte Abstand des Punktes  $p_i = (x_i, y_i)$  zur Geraden  $l_0$  ist. Bei diesem *line-fitting* wird die Summe der Quadrate der senkrechten Abstände der Punkte zur Geraden minimiert. Diese Fehlerfunktion, die angibt wie gut eine Gerade  $l_0$  eine Punktemenge  $P$  beschreibt, ergibt aus geometrischer Sicht Sinn. Dieser Ansatz führt allerdings zu einer nicht-linearen Regression. Für die Lösung dieses Problems wird die Gerade  $l_0 : y = m \cdot x + b$  in Polarkoordinaten  $l_0 = (r, \varphi)$  umgewandelt. Die Abstände  $d(p_i, l_0)$  können in dieser Darstellung der Geraden wie folgt berechnet werden:

$$d(p_i, l_0) = |x_i \cdot \cos \varphi + y_i \cdot \sin \varphi - r|$$

Damit entsteht aus (3.2) die Fehlerfunktion

$$\chi^2(r, \varphi) = \sum_{i=1}^N (x_i \cdot \cos \varphi + y_i \cdot \sin \varphi - r)^2,$$

deren partielle Ableitungen nach  $r$  und  $\varphi$  bei der Minimierung gleich 0 gesetzt werden müssen. Es entsteht ein Gleichungssystem, das durch viele verschiedene Umformungen [Pfister 2002], deren Erläuterung sehr aufwendig ist und hier nicht weiter erklärt werden, zu folgendem Gleichungssystem führt:

$$\tan 2\varphi = \frac{-2 \sum_{i=1}^n (\bar{y} - y_i) \cdot (\bar{x} - x_i)}{\sum_{i=1}^n [(\bar{y} - y_i)^2 - (\bar{x} - x_i)^2]}$$

$$r = \bar{x} \cdot \cos \varphi + \bar{y} \cdot \sin \varphi$$

In diesen Gleichungen sind  $\bar{x}$  und  $\bar{y}$  die Mittelwerte der x- und y-Koordinaten aller Punkte aus  $P$ .

Wir können nun mit diesen Gleichungen eine Gerade an eine Punktmenge anpassen. Vergleichen wir dieses Gleichungssystem mit dem aus *Abschnitt 3.3.1*, dann werden verhältnismäßig gleich viele Operationen für die Berechnung einer Geraden benötigt. Somit ist der Aufwand der nicht-linearen Regression gleich dem Aufwand der linearen Regression. Das Ergebnis im aktuellen Ansatz ist jedoch in geometrischer Hinsicht besser einzuschätzen. In der praktischen Umsetzung des heuristischen Verfahrens haben wir deshalb die nicht-lineare Regression verwendet.

## 3.4 Abbruchkriterium

Bei diesem heuristischen Verfahren zur Erstellung eines Linienmodells werden nacheinander Liniensegmente gebildet, die zusammen am Ende ein Modell der Umgebung darstellen sollen. Das heuristische Verfahren ist beendet, wenn jeder Messpunkt einem Liniensegment zugeordnet wurde. Dieser optimale Fall kann jedoch nur in sehr seltenen Fällen erreicht werden. Dies hängt unter anderem damit zusammen, dass die Umgebung des Roboters auch aus nicht-linearen Strukturen besteht und diese nur schwer durch Liniensegmente darstellbar sind. Messpunkte nicht-linearer Strukturen können somit nur schwer einem Liniensegment zugeordnet werden. Des Weiteren sind in den Messpunkten Ausreißer vorhanden, die gegen Ende des Verfahrens übrig bleiben und nicht durch Liniensegmente erklärt werden können. Vor allem aus diesen zwei Gründen müssen wir ein anderes Abbruchkriterium finden, als darauf zu warten, bis jeder Messpunkte einem Segment zugeordnet wurde.

Ein weiterer Grund, warum nicht alle Messpunkte zugeordnet werden können, hängt auch mit der Filterung der Liniensegmente aus *Abschnitt 3.2.3* zusammen. Dort wird ein Linienmodell nur durch Liniensegmente erweitert, deren Länge größer als *SEG\_LENGTH\_MIN* ist. Entsteht also bei der Segmentierung einer Geraden  $l_0$  ein Liniensegment  $s_0$  das kürzer als *SEG\_LENGTH\_MIN* ist, dann wird  $s_0$  nicht in das derzeitige Linienmodell eingefügt. Der Eintrag in der Akkumulatormatrix, der die Gerade  $l_0$  repräsentiert, wird danach auf 0 gesetzt, damit diese Gerade nicht ein weiteres Mal extrahiert wird. Dies hat zur Folge, dass alle Einträge der Akkumulatormatrix irgendwann gleich 0 sind, auch wenn nicht jeder Messpunkt einem Liniensegment zugeordnet wurde. Wir können also unser heuristisches Verfahren als beendet ansehen, wenn alle Einträge der Akkumulatormatrix gleich 0 sind. Dies kann jedoch sehr lange dauern, da jede Gerade, deren Akkumulatorzelle größer als 0 ist, mindestens einmal aus dem Hough-Raum extrahiert und anschließend segmentiert werden muss.

Wir können schon früher unser Verfahren anhalten, wenn wir die Akkumulatormatrix etwas genauer betrachten. Damit eine Zelle der Akkumulatormatrix auf 0 gesetzt wird, muss zuerst die entsprechende Gerade extrahiert werden. Dies geschieht durch die

Bestimmung der Zelle mit dem maximalen Wert. Angenommen der Wert des Maximums beträgt zwei, dann bedeutet dies, dass nur zwei Punkte kollinear sind und somit auf der Geraden liegen. Die anschließende Segmentierung führt dann mit großer Wahrscheinlichkeit zu einem nicht zulässigen Liniensegment. Wir können dies aber nicht garantieren, da wir nicht wissen, wie viele Messpunkte der Geraden vor der Segmentierung zugeordnet wurden. Das Maximum der Akkumulatormatrix ist trotzdem ein gutes Kriterium, ob wir noch weitere Liniensegmente aus den verbliebenen Messpunkten bilden können. Deshalb führen wir einen Parameter *AKKU\_MIN* ein, der unser Abbruchkriterium festlegt. Ist das Maximum der Akkumulatormatrix kleiner als *AKKU\_MIN*, dann wird das Verfahren angehalten. Die bis dahin erstellte Menge von Liniensegmenten bildet dann unser Linienmodell. Eine in der Praxis geeignete Festlegung des Wertes für *AKKU\_MIN* wird in *Abschnitt 5.2.1* genauer beschrieben.

## 3.5 Bearbeitung der Liniensegmente

In diesem Abschnitt versuchen wir durch drei verschiedene Ansätze die bisher erstellten Linienmodelle zu verbessern. Die Erstellung der Liniensegmente anhand von Messpunkten ist für uns abgeschlossen. Wir betrachten nur noch die Liniensegmente mit ihren zugeordneten Punktemengen.

### 3.5.1 Vereinigung von Liniensegmenten

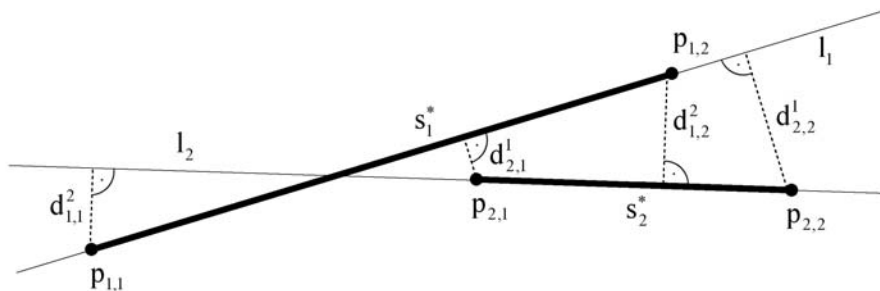
Bei der Erstellung unserer Linienmodelle werden die einzelnen Liniensegmente  $s_i^*$  nacheinander bestimmt. Dadurch entsteht eine Reihenfolge der Liniensegmente, bei der  $s_1^*$  vor  $s_2^*$  vor  $s_3^*$  usw. erstellt wurde. Im Verlauf des Verfahrens kann es vorkommen, dass eine lineare Struktur in unserer Umgebung durch mehrere Liniensegmente beschrieben wird. Angenommen das Liniensegment  $s_j^*$  beschreibt eine lineare Struktur komplett oder nur teilweise. Werden  $s_j^*$  nicht alle Messpunkte der linearen Struktur zugeordnet, dann kann ein weiteres Liniensegment  $s_k^*$ ,  $k > j$ , entstehen, dem die restlichen Punkte der linearen Struktur zugeordnet werden. Das Liniensegment  $s_k^*$  beschreibt somit auch einen Teil der bereits beschriebenen linearen Struktur. Die zwei Liniensegmente  $s_j^*$  und  $s_k^*$  können vereinigt werden, um zum einen die Anzahl der Liniensegmente im unserem Modell zu reduzieren und zum anderen eine lineare Struktur nur durch ein Segment zu beschreiben.

Wir benötigen ein Verfahren zum Entscheiden, ob zwei Liniensegmente vereinigt werden sollen. Die Vereinigung von zwei Liniensegmente wird in anderen Ansätzen der Kartenbildung für Roboter bereits angewendet (z.B. [Pfister et al. 2003] und [Gonzalez et al. 1994]). In der Erstellung von Karten bei Gutmann et al. [2002] werden

Liniensegmente aus einem einzelnen Laserscan gebildet und nicht wie bei uns aus allen Messungen. Nach jedem Scan wird das aktuelle Linienmodell durch die neuen Liniensegmente aktualisiert. Dies führt dazu, dass häufig die neuen Liniensegmente mit anderen aus dem Modell vereinigt werden müssen. Einige Grundideen werden deshalb aus diesen Verfahren übernommen und an unsere jetzige Situation angepasst.

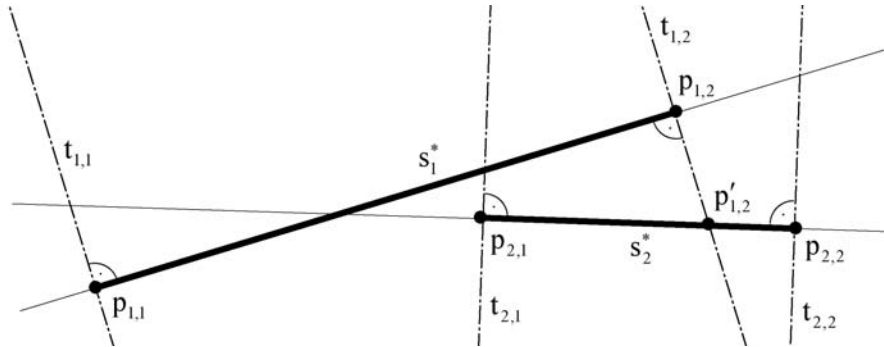
Eine Frage ist nun, zu welchem Zeitpunkt wir die Vereinigung der Liniensegmente überprüfen? Wir könnten gleich nach der Erstellung eines Liniensegmentes  $s_k^*$  prüfen, ob dieses mit einem anderen Liniensegment  $s_j^*$ ,  $j < k$ , vereinigt werden kann. Da die Reihenfolge der Erstellung der Liniensegmente leicht zu merken ist, können wir die Vereinigung der Liniensegmente auch separat zur eigentlichen Erstellung unseres Linienmodells durchführen. Beide Ansätze führen zum gleichen Ergebnis, aber eine separate Behandlung der Vereinigung macht unser gesamtes Verfahren etwas modularer und somit übersichtlicher. In der Praxis erleichtert dies auch ein wenig die Programmierung. Also überprüfen wir die Vereinigung von Liniensegmenten erst nach der kompletten Extraktion von Liniensegmenten. Dabei wird für jedes erstellte Segment  $s_k^*$  überprüft, ob es mit einem  $s_j^*$ ,  $1 \leq j < k$  vereinigt werden kann.

Kommen wir nun zum zentralen Problem der Vereinigung. Damit zwei Liniensegmente vereinigt werden, sollten sie zum einem nah genug beieinander liegen und zum anderen sollten sich die Segmente überlappen. Bevor wir den Algorithmus weiter erklären, definieren wir den Abstand der Endpunkte und die Überlappung von zwei Liniensegmenten. Wir benützen im Folgenden die zwei Liniensegmente  $s_1^*$  und  $s_2^*$ , die durch die Endpunkte  $(p_{1,1}, p_{1,2})$  und  $(p_{2,1}, p_{2,2})$  beschrieben werden, um die Definitionen zu erklären. Die Distanz  $d_i^j$  eines Endpunktes  $p_i$  zur Geraden  $l_j$  des anderen Segments  $s_j^*$  ist gegeben durch den zu  $l_j$  senkrechten Abstand. Dies wird in *Abbildung 3.5* veranschaulicht, in der die Abstände  $d_i^j$  eingezeichnet sind.



**Abbildung 3.5:** Abstände der Endpunkte zu einem anderen Liniensegment

Bei der Definition der Überlappung müssen wir zwischen der Überlappung eines Endpunktes und eines Liniensegments unterscheiden. Betrachten wir zunächst das Beispiel in *Abbildung 3.6* für die Definition der Überlappung eines Endpunktes.



**Abbildung 3.6:** Überlappung der Endpunkte bzw. der Liniensegmente mit einem anderen Liniensegment

Ob ein Endpunkt des Liniensegments  $s_2^*$  mit dem Liniensegment  $s_1^*$  überlappt, wird anhand der zwei Geraden  $t_{1,1}$  und  $t_{1,2}$  entschieden, die senkrecht zu  $s_1^*$  liegen und durch die Endpunkte  $p_{1,1}$  bzw.  $p_{1,2}$  verlaufen. Da  $p_{2,1}$  zwischen den Geraden  $t_{1,1}$  und  $t_{1,2}$  liegt, überlappt  $p_{2,1}$  mit dem Liniensegment  $s_1^*$ . Der Punkt  $p_{2,2}$  hingegen liegt nicht zwischen den zwei Senkrechten und überlappt daher nicht. Um die Überlappung zu beschreiben, führen wir binäre Variablen  $o_i^j$  ein, die wie folgt belegt werden:

$$o_i^j = \begin{cases} 1, & \text{falls } p_i \text{ mit } s_j^* \text{ überlappt} \\ 0, & \text{sonst} \end{cases}$$

In unserem Beispiel ist dann  $o_{2,1}^1 = 1$  und  $o_{2,2}^1 = 0$ . Betrachten wir nun die Lage der Endpunkte  $p_{1,1}$  und  $p_{1,2}$ , dann ist  $o_{1,1}^2 = 0$  und  $o_{1,2}^2 = 1$ .

Dies ist eine etwas andere Definition als bei Gutmann et al. [2002]. In unserem Beispiel (Abbildung 3.6) werden bei der Definition von Gutmann et al. die Geraden  $t_{1,1}$  und  $t_{1,2}$  nicht senkrecht zum Liniensegment  $s_1^*$  gebildet, sondern senkrecht zur x-Achse. Damit müssen nur noch die x-Werte der Endpunkte verglichen werden, um zu entscheiden, ob ein Endpunkt überlappt. Unsere Definition ist zwar aufwendiger, aber in geometrischer Hinsicht nach meiner Meinung nach exakter.

Beim Vergleich eines Liniensegments mit einem anderen ist es nicht nur wichtig zu überprüfen, ob ein Segment überlappt, sondern zu wie viel Prozent es überlappt. Deshalb definieren wir die Überlappung eines Liniensegments  $s_i^*$  mit einem Liniensegment  $s_j^*$  durch den Wert  $P_i^j$  wie folgt:

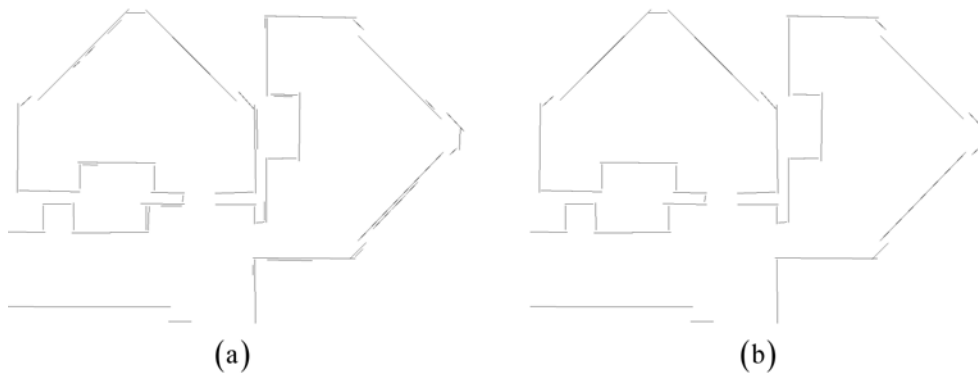
$$P_i^j = \begin{cases} 0 & , \text{falls } o_{i,1}^j = 0 \text{ und } o_{i,2}^j = 0 \\ \frac{d(s_i^*, s_j^*)}{d(p_{i,1}, p_{i,2})} & , \text{falls } o_{i,1}^j = 1 \text{ oder } o_{i,2}^j = 1 \\ 1 & , \text{falls } o_{i,1}^j = 1 \text{ und } o_{i,2}^j = 1 \end{cases}$$

wobei  $d(p_{i,1}, p_{i,2})$  die Länge des Liniensegments  $s_i^*$  angibt und durch den Abstand der beiden Endpunkte  $p_{i,1}$  und  $p_{i,2}$  bestimmt werden kann. Der Term  $d(s_i^*, s_j^*)$  stellt die

Länge der Überlappung von  $s_i^*$  mit  $s_j^*$  dar. Bei unserem Beispiel aus *Abbildung 3.6*, bei dem  $s_2^*$  mit  $s_1^*$  überlappt, ist  $d(s_2^*, s_1^*)$  der Abstand des Punktes  $p_{2,1}$  zum Schnittpunkt  $p'_{1,2}$  und stellt somit die Länge der Überlappung dar. Der Wert  $P_2^j$  gibt dann an, zu wie viel Prozent der Gesamtlänge von  $s_2^*$  das Liniensegment  $s_2^*$  mit  $s_1^*$  überlappt.

Mit den Informationen über die Abstände der Endpunkte zur jeweils anderen Geraden und über die Überlappung können wir nun entscheiden, ob zwei Liniensegmente vereinigt werden. Für die Vereinigung eines Liniensegments  $s_i^*$  mit einem Liniensegment  $s_j^*$  müssen folgende Bedingungen erfüllt werden: zum einen muss  $P_i^j \geq \text{MERGE\_OVERLAP\_MIN}$  sein; zum anderen müssen die Abstände  $d_{i,1}^j$  und  $d_{i,2}^j$  geringer als  $\text{MERGE\_DIST\_MAX}$  sein, damit die Segmente auch nah genug beieinander liegen.

Den Parametern  $\text{MERGE\_OVERLAP\_MIN}$  und  $\text{MERGE\_DIST\_MAX}$  müssen wir zu Beginn der Vereinigung geeignete Werte zuordnen, um eine sinnvolle Vereinigung zu garantieren. Die Festlegung beider Parameter ist nicht sehr schwierig, da bei der Entscheidung, ob zwei Liniensegmente vereinigt werden, nur noch die Liniensegmente betrachtet werden und nicht mehr die Messpunkte. In der Praxis erhalten wir mit den Werten  $\text{MERGE\_OVERLAP\_MIN} = 50\%$  und  $\text{MERGE\_DIST\_MAX} = 10 \text{ cm}$  ein sehr gutes Ergebnis. In *Abbildung 3.7* ist ein Beispiel einer Vereinigung von Liniensegmenten mit diesen Parametern dargestellt.



**Abbildung 3.7:** (a) zeigt ein Linienmodell mit 62 Liniensegmenten vor der Vereinigung der Liniensegmente und (b) ein Linienmodell mit 45 Liniensegmenten danach, wobei die Werte von  $\text{MERGE\_OVERLAP\_MIN}$  mit 50% und  $\text{MERGE\_DIST\_MIN}$  mit 10 cm belegt wurden.

Wird entschieden, dass zwei Liniensegmente vereinigt werden, muss dies geeignet durchgeführt werden. Bei Gutmann et al. [2002] wird das neue Liniensegment anhand der Endpunkte, gewichtet durch die Länge der Liniensegmente, mit einer linearen Regression berechnet, die vergleichbar mit dem *line-fitting* aus *Abschnitt 3.3.1* ist. Wenn die Messungen allgemein gleich dicht verteilt sind, macht dieses Vorgehen auch Sinn und die Berechnung kann relativ schnell durchgeführt werden, da nur die

Endpunkte zu berücksichtigen sind. Bei der Erstellung der einzelnen Liniensegmente kann es durchaus vorkommen, dass den Liniensegmenten relativ zu ihrer Länge unterschiedlich viele Messpunkte zugeordnet werden. Deshalb ist die Gewichtung der Endpunkte durch die Länge der Liniensegmente für unser heuristisches Verfahren nach meiner Meinung nach nicht geeignet, da jeder Messpunkt die Lage eines Liniensegments gleich stark beeinflussen sollte.

Im Vergleich zu Gutmann et al. [2002] haben wir einen wesentlich größeren Aufwand, da wir mehr Punkte berücksichtigen wollen und verwenden deshalb einen etwas anderen Ansatz zur Berechnung des vereinigten Liniensegments. Einem einzelnen Liniensegment wurden bisher Messpunkte zugeordnet, die einer linearen Struktur angehören. Wir konnten jedoch nicht garantieren, dass alle Punkte einer linearen Struktur genau einem Liniensegment zugeordnet wurden. Sollen nun zwei Liniensegmente vereinigt werden, dann verwenden wir die beiden zugeordneten Punktemengen für die Berechnung des neuen Liniensegments. Diese zwei Punktemengen sollten nach Annahme unserer Vereinigung der gleichen linearen Struktur angehören. Wir passen also das neue Liniensegment an die beiden Punktemengen an und verwenden dafür ein *line-fitting* aus *Abschnitt 3.3*. Alle erstellten Liniensegmente unserer Linienmodelle, egal ob aus der Vereinigung entstanden oder nicht, werden mit der gleichen Idee des *line-fittings* an eine Menge von Messpunkten angepasst. Dadurch bleiben wir unserer Idee treu und vermischen nicht verschiedene Ansätze des *line-fittings*.

#### **3.5.2 Veränderung der Länge der Liniensegmente**

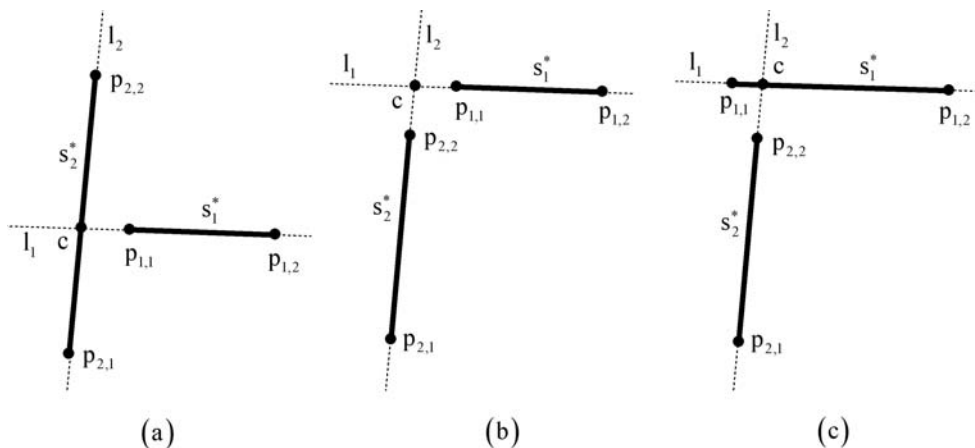
Wir wollen in dieser Arbeit vorwiegend Linienmodelle erstellen, die Innenraumumgebungen beschreiben sollen. Voraussetzung dafür ist, dass die Umgebungen eine Darstellung durch Liniensegmente unterstützen. Dies ist in den meisten Fällen auch gegeben und wir können sogar noch weiter annehmen, dass die linearen Strukturen in den Umgebungen zusammenhängend sind. Für unsere Modelle bedeutet dies, dass ein Liniensegment an einem anderen Liniensegment enden sollte. Betrachten wir das Beispiel in *Abbildung 3.7 (b)*, das ein Linienmodell nach der Vereinigung der Liniensegmente darstellt; darin ist leicht zu erkennen, dass fast alle Liniensegmente nicht an einem anderen Liniensegment enden. Die Liniensegmente sind bei der Segmentierung entweder zu kurz oder zu lang geraten, was mit der zugeordneten Menge von Punkten abhängt. Bei der Zuordnung der Punkte zu einer Geraden in *Abschnitt 3.2.1* gibt der Wert *ASSIGN\_DIST\_MAX* den maximalen Abstand an, den ein Punkt zur Geraden besitzen darf, damit er zur Geraden zugeordnet wird. Entsteht im weiteren Verlauf daraus ein Liniensegment, dann existiert danach kein nicht zugeordneter Messpunkt, der sehr nah an dem Liniensegment liegt. Durch diese Tatsache kann auch kein weiteres Liniensegment gebildet werden, das mit diesem



Liniensegment verbunden ist bzw. sich mit diesem schneidet. Die Abstände der Endpunkte eines Liniensegments zu einem anderen Liniensegment sind zumeist nicht sehr groß, was wiederum mit der Größe von  $ASSIGN\_DIST\_MAX$  zusammenhängt.

In diesem Abschnitt versuchen wir die Länge der Liniensegmente zu verändern, um die Liniensegmente miteinander zu verbinden. Bei der Veränderung der Länge verschieben wir die Endpunkte so, dass sie auf einem anderen Liniensegment oder sogar auf einem anderen Endpunkt liegen, ohne dabei die Ausrichtung der Liniensegmente zu verändern.

Zu jedem Endpunkt muss ein passendes Liniensegment gefunden werden, auf dem der Punkt später liegen soll. Wir betrachten deshalb für jeden Endpunkt eines Liniensegments alle anderen Liniensegmente nacheinander, bis wir ein geeignetes Segment gefunden haben. Es gibt auch Endpunkte, denen kein passendes Segment zugeordnet werden kann, wie zum Beispiel Endpunkte von Liniensegmenten, die eine offene Tür repräsentieren. Diese Endpunkte bleiben zurecht unverändert. Da sich in unserem Linienmodell keine Liniensegmente schneiden, können bei der Suche nach einem geeigneten Liniensegment für die Veränderung eines Endpunktes drei Situationen entstehen, die anhand der Skizzen in *Abbildung 3.8* dargestellt werden. Wie wir uns in diesen Situationen verhalten, wird im Folgenden erläutert:



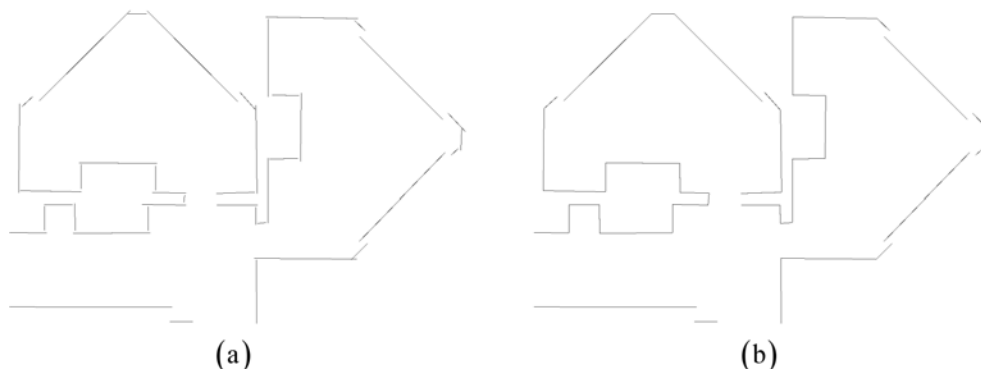
**Abbildung 3.8:** Drei Situationen, die bei der Veränderung der Länge des Liniensegments  $s_i^*$  entstehen können; Der Endpunkt  $p_{i,j}$  des Liniensegments  $s_i^*$  soll jeweils verändert werden.

- (a) Liegt der Schnittpunkt  $c$  der beiden Geraden  $l_1$  und  $l_2$  auf dem Liniensegment  $s_2^*$  wie in *Abbildung 3.8 (a)*, dann kann der Endpunkt  $p_{1,1}$  auf diesen Schnittpunkt verschoben werden. Dadurch verlängern wir das Liniensegment  $s_1^*$  um die Länge  $d(p_{1,1}, c)$ . Dieser Wert ist der Abstand des Punktes  $p_{1,1}$  zu  $c$ . Damit die Verschiebung eines Endpunktes nicht beliebig weit durchgeführt werden kann, beschränken wir die Länge der Veränderung nach oben durch  $EXTEND\_LENGTH\_MAX$ . Ist

also  $d(p_{1,1},c)$  größer  $EXTEND\_LENGTH\_MAX$ , dann darf  $p_{1,1}$  nicht auf das Liniensegment  $s_2^*$  verschoben werden.

- (b) In *Abbildung 3.8 (b)* liegt der Schnittpunkt  $c$  auf keinem der beiden Liniensegmente. Trotzdem kann es sinnvoll sein, die Lage des Endpunktes  $p_{1,1}$  zu verändern. Um dies zu entscheiden, müssen die Abstände  $d(p_{1,1},c)$  und  $d(p_{2,2},c)$  von  $p_{1,1}$  und  $p_{2,2}$  zum Schnittpunkt  $c$  berechnet werden. Sind nun beide Abstände kleiner gleich  $EXTEND\_LENGTH\_MAX$ , dann werden beide Endpunkte  $p_{1,1}$  und  $p_{2,2}$  auf den Schnittpunkt  $c$  gelegt. Wir verlängern beide Liniensegmente, denn eine einzelne Verlängerung würde keinen Sinn ergeben, da ein Endpunkt auf einem anderen Liniensegment liegen soll.
- (c) Wollen wir den Endpunkt  $p_{1,1}$  in *Abbildung 3.8 (c)* auf  $s_2^*$  verschieben, dann fällt auf, dass der Schnittpunkt  $c$  nicht auf  $s_2^*$ , sondern auf  $s_1^*$  liegt. Der Punkt  $p_{1,1}$  wird nur dann auf  $c$  verschoben, wenn auch  $p_{2,2}$  auf  $c$  verschoben wird. Sind also beide Abstände  $d(p_{1,1},c)$  und  $d(p_{2,2},c)$  kleiner gleich  $EXTEND\_LENGTH\_MAX$ , dann wird  $s_1^*$  verkürzt und  $s_2^*$  verlängert. Eine Veränderung von  $p_{1,1}$  allein würde wie in *(b)* keinen Sinn ergeben.

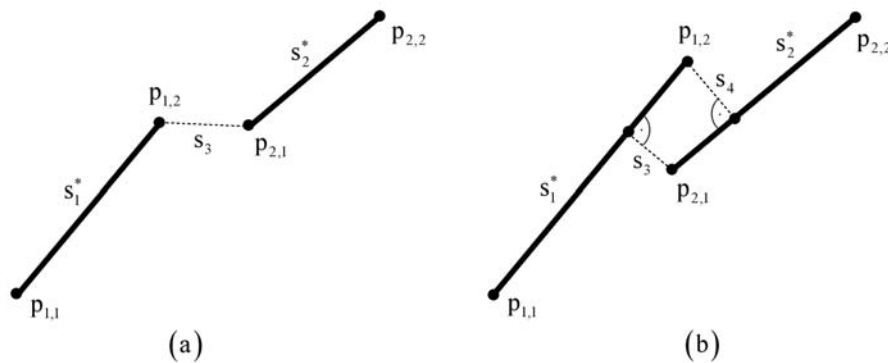
Wie in diesen drei Situationen dargestellt wird entschieden, ob ein Endpunkt auf ein anderes Liniensegment verschoben wird. Dadurch entstehen weitgehend zusammenhängende Linienmodelle, welche die Umgebungen noch besser beschreiben. Bei der Festlegung des Parameter  $EXTEND\_LENGTH\_MAX$  müssen wir einen geeigneten Wert finden, damit die Veränderung der Länge nicht zu fehlerhaften Liniensegmenten führt. Wie in *Abbildung 3.9 (a)* zu erkennen ist, müssen die Liniensegmente nur sehr gering verändert werden. In der Praxis haben sich Werte zwischen 15 cm und 25 cm für  $EXTEND\_LENGTH\_MAX$  bewährt. Die *Abbildung 3.9 (b)* zeigt ein Linienmodell nach Veränderung der Länge der Liniensegmente mit  $EXTEND\_LENGTH\_MAX = 20$  cm.



**Abbildung 3.9:** (a) zeigt ein Linienmodell vor der Veränderung der Länge der Liniensegmente und (b) danach, wobei  $EXTEND\_LENGTH\_MAX$  mit einem Wert von 20 cm belegt wurde.

### 3.5.3 Einfügen „kurzer“ Liniensegmente

Durch die Filterung der Liniensegmente in *Abschnitt 3.2.3* können keine Segmente mit einer kürzeren Länge als  $SEG\_LENGTH\_MIN$  entstehen. Dadurch kann nicht jeder Endpunkt eines Liniensegments, wie im Abschnitt zuvor erklärt, auf ein anderes Liniensegment verschoben werden, damit am Ende die Liniensegmente miteinander verbunden sind. Liegen zum Beispiel zwei Liniensegmente annähernd parallel zueinander, dann kann keine Veränderung der Länge der Segmente durchgeführt werden, auch wenn sie nur wenige Zentimeter voneinander entfernt sind. In einigen Situationen wäre es trotzdem angebracht, diese zwei Liniensegmente zu verbinden. In *Abbildung 3.10* sind zwei solche Situationen dargestellt, in denen zwei Liniensegmente verbunden werden sollen.



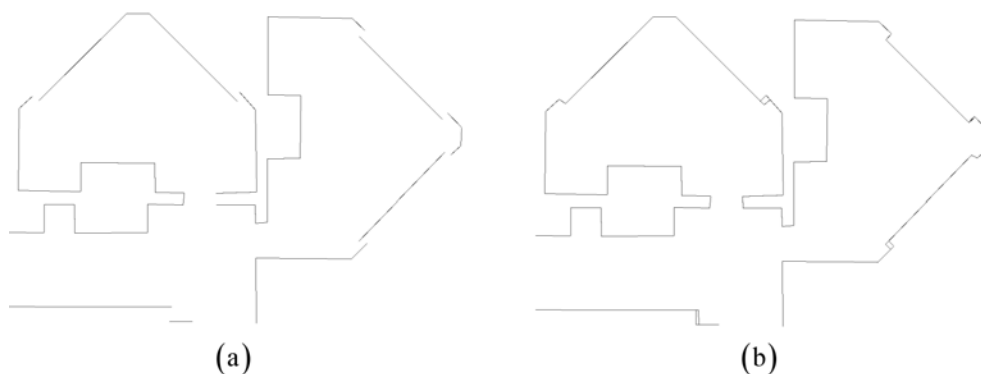
**Abbildung 3.10:** Beide Skizzen stellen jeweils zwei Liniensegmente dar, welche bei (a) durch ein und bei (b) durch zwei neue „kurze“ Liniensegmente verbunden werden sollen.

Betrachten wir zuerst den etwas einfacheren Fall in *Abbildung 3.10 (a)*, bei dem sich die Liniensegmente nicht überlappen. Ist der Abstand der Endpunkte  $p_{1,2}$  und  $p_{2,1}$  kleiner als  $SHORT\_SEGMENT\_MAX$ , dann fügen wir dem Linienmodell ein neues Liniensegment  $s_3$  hinzu, das durch die Endpunkte  $p_{1,2}$  und  $p_{2,1}$  beschrieben wird. Dadurch verbinden wir  $s_1^*$  und  $s_2^*$  durch ein neues Liniensegment, dem keine Messpunkte zugeordnet werden. Natürlich darf  $SHORT\_SEGMENT\_MAX$  nicht zu groß gewählt werden, da ansonsten Segmente entstehen können, die nicht sinnvoll sind. Bei der Festlegung dieses Wertes kann man sich zum Beispiel an der Breite des Roboters orientieren, wenn das Linienmodell für die Navigation des Roboters eingesetzt werden soll. Dann macht das Einfügen des neuen Segments auch Sinn, da der Roboter durch die schmale Lücke zwischen den beiden Liniensegmente  $s_1^*$  und  $s_2^*$  sowieso nicht hindurchfahren kann.

Kommen wir nun zur *Abbildung 3.10 (b)*, in der sich die zwei Liniensegmente überlappen. Solch eine Situation kann häufiger in der Praxis auftreten, wie zum

Beispiel bei einer geschlossenen Tür. Die linearen Strukturen der Wand und der Tür sind zumeist parallel und nur durch eine kleine senkrechte Struktur miteinander verbunden. Diese kleine Struktur wird bei der Erstellung von Liniensegmenten jedoch nicht gefunden, da sie zu klein ist oder ein Großteil ihrer Messpunkte bereits Liniensegmenten der Wand und der Tür zugeordnet wurden. Um nun die zwei Segmente  $s_1^*$  und  $s_2^*$  zu verbinden, fügen wir die zwei neuen Liniensegmente  $s_3$  und  $s_4$  ein. Wir könnten, wie in der Situation zuvor, nur ein neues Liniensegment einfügen, das durch die beiden Endpunkte  $p_{1,2}$  und  $p_{2,1}$  beschrieben wird. Aber nicht in jeder Situation liegen diese zwei Endpunkte nah genug beieinander. Deshalb wird an beiden Endpunkten  $p_{1,2}$  und  $p_{2,1}$  jeweils ein neues Liniensegment eingefügt, deren Länge nicht größer als  $SHORT\_SEGMENT\_MAX$  sein darf.

Bei der Festlegung des Wertes für  $SHORT\_SEGMENT\_MAX$  orientieren wir uns an üblichen Maßen von Robotern. Da der Parameter auf keinen Fall größer als die Breite eines Roboters sein darf, eignen sich Werte zwischen 25 cm und 50 cm. In *Abbildung 3.11* ist ein Linienmodell vor und nach dem Einfügen „kurzer“ Liniensegmente dargestellt, wobei  $SHORT\_SEGMENT\_MAX = 35$  cm verwendet wurde.

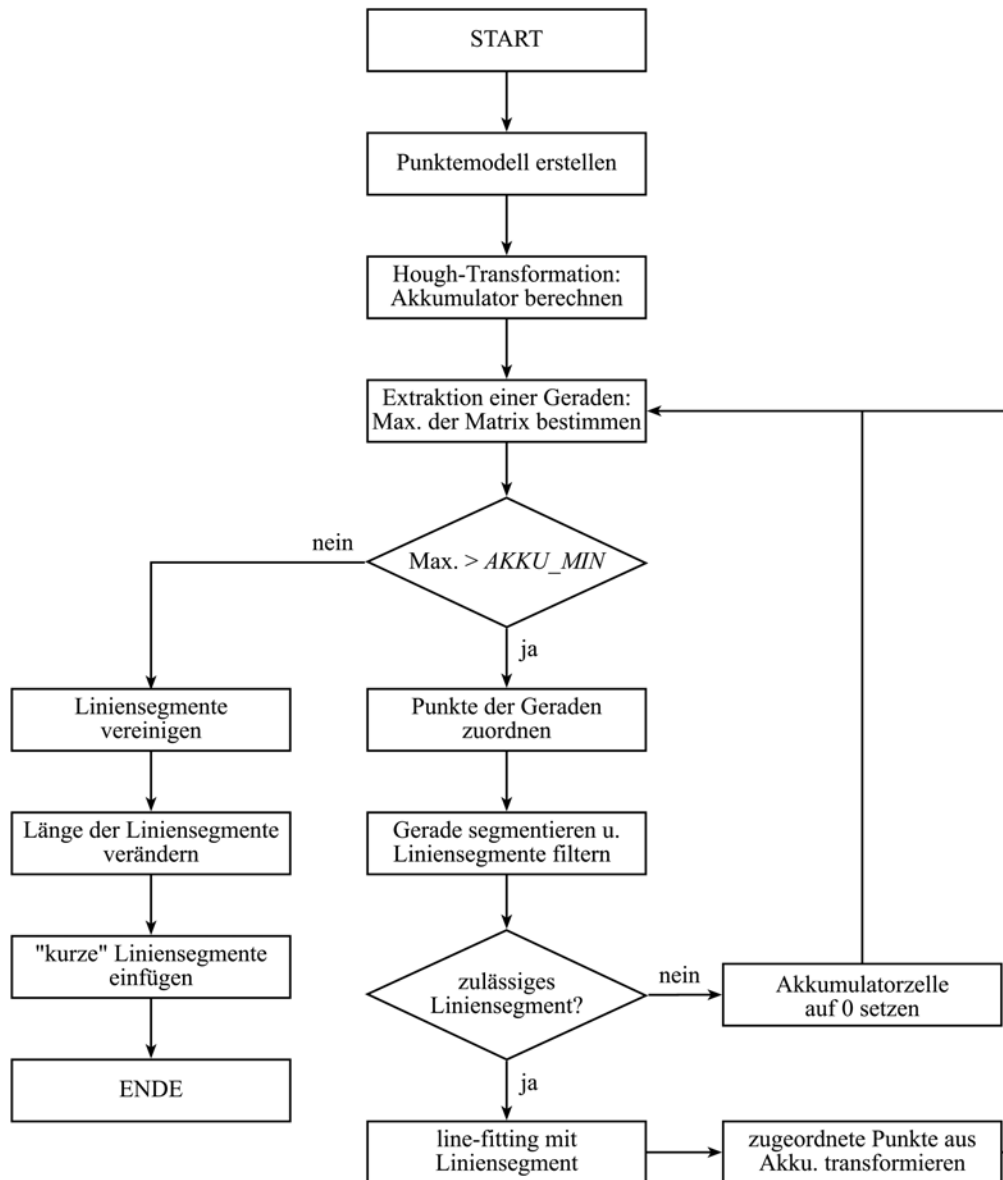


**Abbildung 3.11:** (a) zeigt ein Linienmodell vor dem Einfügen „kurzer“ Liniensegmente und (b) danach, wobei  $SHORT\_SEGMENT\_MAX$  mit einem Wert von 35 cm belegt wurde.

Nachdem wir die Veränderung der Länge der Liniensegmente und das Einfügen „kurzer“ Liniensegmente für noch nicht verbundene Endpunkte durchgeführt haben, ist fast jeder Endpunkt eines Liniensegments mit einem anderen Liniensegment verbunden. Wir erhalten dadurch ein Linienmodell, deren Liniensegmente zusammenhängend sind und nur dann fehlerhafte Lücken aufweist, wenn die Exploration der Umgebung zu ungenau war.

### 3.6 Flussdiagramm des Verfahrens

Das gesamte heuristische Verfahren zur Erstellung von Linienmodellen wird durch ein Flussdiagramm in *Abbildung 3.12* visualisiert, um einen zusammenfassenden Überblick zu geben.



**Abbildung 3.12:** Das Flussdiagramm zeigt den Ablauf der wichtigsten Schritte des gesamten heuristischen Verfahrens zur Erstellung von Linienmodellen.



## 4 Lernen von Linienmodellen mit EM

In diesem Kapitel wird ein weiteres Verfahren zur Generierung von Linienmodellen von Umgebungen mobiler Roboter beschrieben, welches in dieser Arbeit Anwendung findet. Das generelle Problem ist, eine Menge von Liniensegmenten zu finden, welche die Messpunkte eines Punktemodells am besten erklären.

Bei der Exploration der Umgebung werden nicht immer alle Hindernisse komplett und lückenlos gescannt. Dazu kommt, dass die erstellten Messpunkten mit einer gewissen Wahrscheinlichkeit von ihren tatsächlichen Positionen abweichen, bedingt unter anderem durch Messfehler und Selbstlokalisierung. Dies erschwert unsere Aufgabe der Erstellung eines Linienmodells wesentlich. Wir verwenden nun ein Verfahren zur Erstellung eines Linienmodells, das die Wahrscheinlichkeit der korrekten Lage unserer Messpunkte maximieren soll. Hierfür wird eine Variante der Erwartungsmaximierung, auch EM (*expectation maximization*) genannt, verwendet. Dieser EM-Ansatz verbindet das Suchen nach einem kompakten Linienmodell und die Zuordnung von Messpunkten zu optimal ausgerichteten Geraden.

EM-Verfahren werden im Allgemeinen für die Lösung von Optimierungsproblemen eingesetzt. In der Informatik, und vor allem auch in der Robotik, gibt es verschiedene Verwendungsmöglichkeiten dieses Ansatzes. Wir haben uns bei der Entwicklung unseres Verfahrens an den in der Praxis erfolgreichen Arbeiten von Hähnel et al. [2001], Martin und Thrun [2002] und Bennewitz et al. [2002] orientiert und einige mathematische Grundlagen übernommen.

Die Beschreibung des EM-Verfahrens in diesem Kapitel beginnt mit der mathematischen Definition der Modelle und der Wahrscheinlichkeitsfunktion. Anschließend folgt die Erläuterung der Erwartungsmaximierung. Hierbei ist zu beachten, dass bei der Durchführung des Verfahrens bis einschließlich zur Erwartungsmaximierung keine Liniensegmente verwendet werden, sondern Geraden. Dies vereinfacht unseren Algorithmus sehr, da eine Behandlung von Liniensegmenten eine Menge weiterer Probleme mit sich bringen würde. Erst nach der Maximierung werden die Geraden in Liniensegmente umgewandelt. Das erstellte Linienmodell wird zum Schluss durch einige kleine Veränderungen verbessert.

## 4.1 Geraden- und Punktemodell

In unserem EM-Verfahren erstellen wir zunächst anhand eines gegebenen Punktemodells ein Geradenmodell, das aus einer endlichen Menge von Geraden besteht. Wir bezeichnen das gesamte Geradenmodell mit  $\theta$ , die Anzahl der Geraden im Modell mit  $J$  und jede individuelle Gerade des Modells mit  $\theta_j$ . Daraus folgt:

$$\theta = \{\theta_1, \theta_2, \dots, \theta_J\}$$

Eine Gerade  $\theta_j$  wird durch zwei Variablen  $(r_j, \varphi_j)$  in Polarkoordinaten beschrieben (siehe *Abschnitt 2.2.1*). Mit dieser Darstellung kann der euklidische Abstand  $d(p, \theta_j)$  eines Punkt  $p = (x, y)$  zu einer Geraden  $\theta_j$  wie folgt berechnet werden:

$$d(p, \theta_j) = |x \cdot \cos \varphi_j + y \cdot \sin \varphi_j - r_j|$$

Jede Messung eines Laserscanners wird zu einem Punkt  $z_i \in \mathfrak{R}^2$  im kartesischen Koordinatensystem umgewandelt. Die Messungen werden zunächst gefiltert, wie in *Abschnitt 2.1.3* und *2.2.3* beschrieben, damit zum Beispiel Ausreißer und dynamische Hindernisse in den Messungen später das Ergebnis nicht zu sehr beeinflussen. Wir bezeichnen die übrig gebliebene Menge aller Messungen mit

$$Z = \{z_i\}.$$

Ein Modell der Messungen, auch Punktemodell genannt, verbindet die Beziehung zwischen einem Geradenmodell und den Messungen  $Z$ . Ein Punktemodell ist ein auf Wahrscheinlichkeiten beruhendes Modell, gegeben einer Umgebung. Angenommen ein Geradenmodell  $\theta$  beschreibt eine Umgebung, dann kann jedem Messpunkt  $z_i$  eine bedingte Wahrscheinlichkeit über seine Korrektheit zugeordnet werden:

$$p(z_i | \theta)$$

Wir nehmen in diesem Ansatz einen gauss'schen Fehler in den Messungen an. Sei nun  $j$  der Index einer Geraden, die den kürzesten Abstand zu einer Messung  $z_i$  hat, dann ist die Fehlerverteilung durch folgende Normalverteilung mit Varianzparameter  $\sigma^5$  gegeben:

$$p(z_i | \theta_j) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \frac{d^2(z_i, \theta_j)}{\sigma^2}} \quad (4.1)$$

---

<sup>5</sup> Alle in diesen Kapitel benötigten Parameter sind im Anhang zusammenfassend aufgelistet und werden dort kurz beschrieben (Seite 113).



## 4.2 Die Wahrscheinlichkeitsfunktion

In der Statistik ist das Finden von glaubwürdigen Modellen gleichbedeutend mit der Maximierung einer Wahrscheinlichkeitsfunktion. Um eine Wahrscheinlichkeitsfunktion für ein Punktemodell zu definieren, führen wir eine Menge von Korrespondenzvariablen  $c_{ij}$  ein, auch Zuordnungsvariablen genannt. Dabei ist  $i$  der Index einer Messung  $z_i$  und  $j$  der Index einer Geraden  $\theta_j$ . Jede Zuordnung  $c_{ij}$  ist eine binäre Variable, die 1 oder 0 ist. Für die Variable  $c_{ij}$  gilt:

$$c_{ij} = \begin{cases} 1, & \text{falls } z_i \text{ zu } \theta_j \text{ zugeordnet wird} \\ 0, & \text{sonst} \end{cases} \quad (4.2)$$

Der Zuordnungsvektor  $C_i$  eines Punktes  $z_i$  ist gegeben durch

$$C_i = \{c_{i1}, c_{i2}, \dots, c_{iJ}\}.$$

Jeder Punkt kann nur genau einer Geraden zugeordnet werden; somit ist nur genau eine Zuordnungsvariable eines Vektors  $C_i$  gleich 1 und daraus folgt:

$$\sum_{j=1}^J c_{ij} = 1, \quad \forall i \quad (4.3)$$

Wenn wir ein Geradenmodell  $\theta$  und alle Zuordnungen  $C_i$  kennen, dann können wir die bedingte Wahrscheinlichkeit eines Punktes  $z_i$  folgendermaßen beschreiben:

$$p(z_i | C_i, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2}} \quad (4.4)$$

Dies verallgemeinert unsere Definition (4.1), da für jeden Punkt  $z_i$  nur eine einzige Zuordnung  $c_{ij}$  gleich 1 ist; alle anderen Zuordnungen sind 0. Belegen wir nun alle Zuordnungsvariablen mit einem Wert, unter Berücksichtigung von (4.2) und (4.3), dann erhalten wir folgende Wahrscheinlichkeiten für die Messungen  $z_i$  zusammen mit ihren Zuordnungsvariablen  $C_i$ :

$$p(z_i, C_i | \theta) = \frac{1}{J \cdot \sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2}} \quad (4.5)$$

In dieser Formel wird angenommen, dass alle  $J$  Zuordnungsvariablen gleich wahrscheinlich sind, ohne Berücksichtigung der Messungen, weshalb  $J$  in den Nenner geschrieben wird. Da wir nun alle Wahrscheinlichkeiten der einzelnen Messungen mit ihren Zuordnungsvariablen kennen, können wir die Wahrscheinlichkeiten aller Messungen mit ihren Zuordnungen zusammen beschreiben. Hierfür nehmen wir an, dass die Fehler in den Messungen untereinander unabhängig sind. Für alle Messungen  $Z = \{z_i\}$  und ihre Zuordnungen  $C = \{c_{ij}\}$  gilt:

$$p(Z, C | \theta) = \prod_i \frac{1}{J \cdot \sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2}} \quad (4.6)$$

Diese Gleichung stellt das Produkt von (4.5) über alle Messungen dar. Im nächsten Abschnitt soll diese Funktion maximiert werden. Anstatt genau diese Funktion zu maximieren, ist es in der Praxis üblich, die logarithmische Wahrscheinlichkeitsfunktion zu maximieren, welche gegeben ist durch:

$$\ln p(Z, C | \theta) = \sum_i \left[ \ln \frac{1}{J \cdot \sqrt{2\pi\sigma^2}} - \frac{1}{2} \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2} \right] \quad (4.7)$$

Der Vorteil der logarithmischen Funktion (4.7) ist die beinhaltete Summe, gegenüber dem Produkt in (4.6). Die Maximierung von  $\ln p(Z, C | \theta)$  kann nun bequemer durchgeführt werden. Das Ergebnis bleibt jedoch gleich, da der Logarithmus streng monoton ist.

Wir sind eigentlich nur an den Modellparametern für ein Geradenmodell  $\theta$  interessiert. Die Zuordnungen  $C$  hingegen bestimmen nur das wahrscheinlichste Modell. Da diese Zuordnungen anfangs nicht bestimmbar sind, ist der normale Weg eine Integration über  $C$ . Anstatt der Integration kann man den Wert der Erwartung der logarithmischen Wahrscheinlichkeitsfunktion (4.7) maximieren, wobei die Erwartung über alle  $C$  genommen wird [Burgard et al. 2001]. Dieser Wert, bezeichnet mit  $E_c [\ln p(Z, C | \theta) | \theta, Z]$ , beschreibt die Erwartung der logarithmischen Wahrscheinlichkeit der Messungen  $Z$ , gegeben das Modell  $\theta$ . Diese Erwartung kann direkt aus (4.7) gewonnen werden.

$$E_c [\ln p(Z, C | \theta) | \theta, Z] = E_c \left[ \sum_i \ln \frac{1}{J \cdot \sqrt{2\pi\sigma^2}} - \frac{1}{2} \sum_j c_{ij} \cdot \frac{d^2(z_i, \theta_j)}{\sigma^2} \right] \quad (4.8)$$

Da die Erwartung ein linearer Operator ist, kann (4.8) wie folgt umgeschrieben werden:

$$E_c [\ln p(Z, C | \theta) | \theta, Z] = \sum_i \left[ \ln \frac{1}{J \cdot \sqrt{2\pi\sigma^2}} - \frac{1}{2} \sum_j E[c_{ij} | \theta, Z] \cdot \frac{d^2(z_i, \theta_j)}{\sigma^2} \right] \quad (4.9)$$

Diese Gleichung bildet die Grundlage des EM-Verfahrens, und wir versuchen sie im folgenden Abschnitt zu maximieren.

### 4.3 Erwartungsmaximierung

In diesem Abschnitt soll nun  $E_c$  aus (4.9) maximiert werden. EM ist ein iterativer Algorithmus, der in jeder Iteration ein Modell mit größerer Erwartung der logarithmischen Wahrscheinlichkeit als beim vorangegangenen Modell erstellt. Anders

gesagt, es entsteht eine Sequenz von Modellen  $\theta^{[1]}, \theta^{[2]}, \theta^{[3]}, \dots$  mit zunehmender Wahrscheinlichkeit, bis die Erwartung konvergiert. Dies kann somit auch als ein klassisches *hill-climbing*-Verfahren in einem Wahrscheinlichkeitsraum angesehen werden. Um mathematisch korrekt zu bleiben, verwandeln wir die Funktion (4.8) in eine sogenannte  $Q$ -Funktion, die von den zwei Modellen  $\theta$  und  $\theta'$  abhängt.

$$Q(\theta'|\theta) = \sum_i \left[ \ln \frac{1}{J \cdot \sqrt{2\pi\sigma^2}} - \frac{1}{2} \sum_j E[c_{ij}|\theta, Z] \cdot \frac{d^2(z_i, \theta'_j)}{\sigma^2} \right] \quad (4.10)$$

Eine Sequenz von Modellen wird berechnet durch

$$\theta^{[v+1]} = \arg \max_{\theta'} Q(\theta'|\theta^{[v]}), \quad (4.11)$$

wobei mit einem Modell  $\theta^{[0]}$  begonnen wird. Da in unserem Falle die  $Q$ -Funktion stetig ist, erhalten wir bei Konvergenz ein lokales Maximum. Das Ergebnis ist ein lokales und somit oftmals kein globales Maximum, welches ein generelles Problem bei *hill-climbing*-Verfahren ist. Die Wahl des Modells  $\theta^{[0]}$  ist von großer Bedeutung, da diese Wahl das Ergebnis sehr stark beeinflussen kann. Wie wir ein gutes Modell  $\theta^{[0]}$  erhalten und welche Probleme damit verbunden sind, wird im *Abschnitt 4.4* näher beschrieben.

Jedes neue Modell  $\theta^{[v+1]}$  erhalten wir durch die Ausführung zweier Schritte:

- *E-Schritt (expectation step)*: Berechnung aller Erwartungen  $E[c_{ij}|\theta^{[v]}, Z]$  für die unbekanntenen Zuordnungen, gegeben das Modell  $\theta^{[v]}$  und die Messpunkte  $Z$ .
- *M-Schritt (maximization step)*: Finden eines neuen Modells  $\theta^{[v+1]}$ , welches die Erwartung der logarithmischen Wahrscheinlichkeit der Messpunkte maximiert, unter Berücksichtigung der Erwartungen aus dem E-Schritt.

### 4.3.1 Der E-Schritt

Anhand eines gegebenen Modells  $\theta^{[v]}$  und der Messungen  $Z$  bestimmen wir im E-Schritt die Erwartungen  $E[c_{ij}|\theta^{[v]}, Z]$  für alle  $i, j$ . Wir können zunächst die Erwartungen wie folgt umformen:

$$\begin{aligned} E[c_{ij}|\theta^{[v]}, Z] &= p(c_{ij}|\theta^{[v]}, Z) \\ &= \sum_{c=0}^l c \cdot p(c_{ij} = c|\theta^{[v]}, Z) \\ &= p(c_{ij} = l|\theta^{[v]}, Z) \\ &= p(c_{ij}|\theta^{[v]}, z_i) \end{aligned}$$

Unter Annahme der Unabhängigkeit der einzelnen Messungen untereinander, können wir, mit Hilfe der Bayes-Regel, die gewünschten Erwartungen berechnen:

$$\begin{aligned}
 E[c_{ij} | \theta^{[v]}, Z] &= p(c_{ij} | \theta^{[v]}, z_i) \\
 &= \frac{p(z_i | \theta^{[v]}, c_{ij}) \cdot p(c_{ij} | \theta^{[v]})}{p(z_i | \theta^{[v]})} \\
 &= \frac{p(z_i | \theta^{[v]}, c_{ij}) \cdot p(c_{ij} | \theta^{[v]})}{\sum_k p(z_i | \theta^{[v]}, c_{ik}) \cdot p(c_{ik} | \theta^{[v]})} \\
 &= \frac{p(z_i | \theta^{[v]}, c_{ij})}{\sum_k p(z_i | \theta^{[v]}, c_{ik})}
 \end{aligned}$$

Im letzten Schritt dieser Umformung wird angenommen, dass  $p(c_{ij} | \theta^{[v]})$  und  $p(c_{ik} | \theta^{[v]})$  für alle  $k$  gleich sind.

Nun müssen nur noch alle  $p(z_i | \theta^{[v]}, c_{ij})$  bestimmt werden, um die Erwartungen zu berechnen. In dem Ansatz aus (4.4) wird lediglich die Distanz eines Punktes zu einer Geraden berücksichtigt. Um die Erwartung nicht allein von der Distanz abhängig zu machen, könnten wir zusätzlich die Differenz zwischen der Ausrichtung einer Geraden und der lokalen Ausrichtung eines Punktes berücksichtigen, um das Ergebnis unter Umständen zu verbessern. Wir unterscheiden im Folgenden zwischen zwei Varianten zur Berechnung der Erwartungen:

#### **Berücksichtigung (nur) der Distanz**

Wir berücksichtigen hier nur den Abstand eines Punktes zu einer Geraden, um die Erwartungen zu berechnen.

$$E[c_{ij} | \theta^{[v]}, Z] = \frac{e^{-\frac{1}{2} \frac{d^2(z_i, \theta_j^{[v]})}{\sigma^2}}}{\sum_k e^{-\frac{1}{2} \frac{d^2(z_i, \theta_k^{[v]})}{\sigma^2}}}$$

Damit haben wir die Erwartung, dass eine Messung  $z_i$  einer Geraden  $\theta_j^{[v]}$  zugeordnet wird, proportional zur *Mahalanobis-Distanz* zwischen den Geraden und dem Punkt berechnet. Die Summe der Erwartungen über alle  $j$  bei einem gegebenen  $i$  ist 1.

In der praktischen Umsetzung müssen für einen Punkt alle Abstände zu den Geraden im momentanen Modell berechnet werden. Anschließend können damit die Erwartungen berechnet werden.

#### **Berücksichtigung der Distanz und der Ausrichtung**

Hier wird zusätzlich zur Distanz die Differenz zwischen der lokalen Ausrichtung eines Punktes und der Ausrichtung einer Geraden berücksichtigt. Die Grundidee dieser

Erweiterung ist, dass ein Punkt mit ähnlicher Ausrichtung zu einer Geraden eine höhere Erwartung besitzt, dieser Geraden zugeordnet zu werden, als ein Punkt mit unterschiedlicherer Ausrichtung. Auf diese Weise reicht es nicht allein aus, dass ein Punkt nah genug an einer Geraden liegt, um mit einer hohen Erwartung dieser Geraden zugeordnet zu werden. Die Differenz zwischen der lokalen Ausrichtung  $\omega_i$  eines Punktes  $z_i$  und der Ausrichtung  $\psi_j$  einer Geraden  $\theta_j$  wird definiert als

$$\Delta a(z_i, \theta_j) = \begin{cases} |\omega_i - \psi_j|, & \text{falls } |\omega_i - \psi_j| \leq \frac{\pi}{2} \\ \pi - |\omega_i - \psi_j|, & \text{sonst} \end{cases}$$

Diese Fallunterscheidung muss durchgeführt werden, da die maximale Differenz zwischen den beiden Ausrichtungen nur  $\pi/2$  betragen kann. Dies ist der Fall, wenn beide Ausrichtungen senkrecht zueinander stehen. Die Erwartungen unter Berücksichtigung der Distanz und der Ausrichtung werden mit  $E^*[c_{ij} | \theta^{[v]}, Z]$  bezeichnet und werden folgendermaßen berechnet:

$$E^*[c_{ij} | \theta^{[v]}, Z] = \frac{e^{-\frac{1}{2} \frac{d^2(z_i, \theta_j^{[v]})}{\sigma^2} - \frac{\Delta a^2(z_i, \theta_j^{[v]})}{\sigma^2}}}{\sum_k e^{-\frac{1}{2} \frac{d^2(z_i, \theta_k^{[v]})}{\sigma^2} - \frac{\Delta a^2(z_i, \theta_k^{[v]})}{\sigma^2}}}$$

Die Summe der Erwartungen über alle  $j$  bei einem gegebenen  $i$  ist wiederum 1.

In der Praxis müssen für einen Punkte alle Differenzen der Ausrichtung und alle Abstände zu den Geraden im momentanen Modell berechnet werden.

### 4.3.2 Der M-Schritt

Nachdem wir die Erwartungen im E-Schritt berechnet haben, können wir das Modell  $\theta^{[v+1]}$  berechnen, dass die Erwartung der logarithmischen Wahrscheinlichkeit der Messungen maximiert. Wir suchen die Werte der Parameter  $(r_j, \varphi_j)$  der Geraden  $\theta_j^{[v+1]}$ , um das neue Modell  $\theta^{[v+1]}$  zu erhalten.

Da einige Terme in der  $Q$ -Funktion (4.10) nicht von den Variablen des Modells  $\theta$  abhängig sind, können wir diese Funktion vereinfachen und es entsteht aus der Maximierung (4.11) die folgende Minimierung:

$$\theta^{[v+1]} = \arg \min_{\theta'} \sum_i \sum_j E[c_{ij} | \theta^{[v]}, Z] \cdot d^2(z_i, \theta'_j) \quad (4.12)$$

Um nun die Minimierung der Gleichung (4.12) durchzuführen, berechnen wir jede Gerade des Modells  $\theta^{[v]}$  neu. Für jede einzelne Gerade  $\theta_j^{[v]}$  wissen wir nach dem E-Schritt, mit welchen Erwartungen die Punkte  $z_i$  dieser Geraden zugeordnet werden.

Um nun die neuen Werte jeder Geraden  $\theta_j^{[v+1]}$  zu erhalten, müssen die partiellen Ableitungen des Terms

$$\sum_i \sum_j E[c_{ij} | \theta^{[v]}, Z] \cdot d^2(z_i, \theta_j') \quad (4.13)$$

nach allen Variablen  $(r_j, \varphi_j)$  gebildet werden. Dies ist eine etwas aufwendigere Rechnung (vgl. [Pfister 2002]), da im Term (4.13) das Quadrat der Abstandsfunktion  $d(z_i, \theta_j') = |x_i \cdot \cos \varphi_j' + y_i \cdot \sin \varphi_j' - r_j'|$  enthalten ist. Das Ergebnis führt zu einer gewichteten *line-fitting*-Formel, die wir bereits in ähnlicher Form in *Abschnitt 3.3.2* kennen gelernt haben. Wir können die Parameter  $(r_j, \varphi_j)$  jeder einzelnen Geraden  $\theta_j^{[v+1]}$  folgendermaßen berechnen:

$$\tan 2\varphi_j = \frac{-2 \sum_i E[c_{ij} | \theta^{[v]}, Z] \cdot (\bar{y} - y_i) \cdot (\bar{x} - x_i)}{\sum_i E[c_{ij} | \theta^{[v]}, Z] \cdot [(\bar{y} - y_i)^2 - (\bar{x} - x_i)^2]}$$

$$r_j = \bar{x} \cdot \cos \varphi_j + \bar{y} \cdot \sin \varphi_j$$

In diesen Gleichungen sind  $\bar{x}$  und  $\bar{y}$  die mit  $E[c_{ij} | \theta^{[v]}, Z]$  gewichteten Mittel der x- und y-Koordinaten aller Punkte  $z_i$  und werden wie folgt berechnet:

$$\bar{x} = \frac{I}{\sum_i E[c_{ij} | \theta^{[v]}, Z]} \cdot \sum_i E[c_{ij} | \theta^{[v]}, Z] \cdot x_i$$

$$\bar{y} = \frac{I}{\sum_i E[c_{ij} | \theta^{[v]}, Z]} \cdot \sum_i E[c_{ij} | \theta^{[v]}, Z] \cdot y_i$$

Jede einzelne Gerade  $\theta_j^{[v+1]}$  wird somit an die mit den Erwartungen gewichteten Punkte angepasst. Wir erhalten dadurch das neue Modell  $\theta^{[v+1]}$ , das die Wahrscheinlichkeit der Messungen maximiert, unter Berücksichtigung aller Erwartungen  $E[c_{ij} | \theta^{[v]}, Z]$ .

## 4.4 Bestimmung der Anzahl von Geraden

Ein generelles Problem aller auf EM basierenden Verfahren ist die Festlegung des Startmodells  $\theta^{[0]}$  zu Beginn der Iteration der Erwartungsmaximierung. In unserem Fall wissen wir im Voraus nicht, welches die optimale Anzahl  $J$  von Geraden  $\theta_j$  ist, um unsere Messpunkte bestmöglich zu beschreiben. Einerseits könnten wir zu Beginn die Anzahl  $J$  auf einen sehr großen Wert setzen. Dies hätte zur Folge, dass Punkte durch mehrere verschiedene Geraden gut erklärt werden könnten. Im Extremfall würde jeder Geraden genau ein Punkt zugeordnet. Deshalb darf die Anzahl der Geraden nicht zu groß gewählt werden, da sonst ein sogenanntes *overfitting* entstehen würde, bei dem die

Wahrscheinlichkeit zwar sehr hoch, aber das Gesamtergebnis nicht akzeptabel ist. Andererseits könnten wir eine geringe Anzahl von Geraden annehmen. Existieren jedoch mehr lineare Strukturen in einer Umgebung als festgelegte Geraden, dann würde eine große Menge von Punkten nicht gut genug erklärt werden. Somit würde die Gesamtwahrscheinlichkeit der Punkte, die durch das Geradenmodell erklärt werden sollen, einen geringen Wert besitzen. Deshalb bestimmen wir die optimale Anzahl  $J$  von Geraden  $\theta_j$  für unser Modell  $\theta$  im Laufe des Lernprozesses.

Nachdem der EM-Algorithmus für ein Modell konvergiert, versuchen wir dieses Modell zu verbessern, indem wir die Anzahl der Geraden vergrößern oder verkleinern. Um dies zu entscheiden, beobachten wir im Laufe der Optimierung die folgenden zwei Sachverhalte:

- *niedrige Wahrscheinlichkeit der Punkte:* Besitzen einige Punkt  $z_i$  eine niedrige Wahrscheinlichkeit im Modell  $\theta$  - d.h. der geringste Abstand der einzelnen Punkte zu einer Geraden ist nicht klein genug - dann ist dies ein Zeichen dafür, dass zu wenig Geraden in diesem Modell vorhanden sind.
- *niedriger Nutzen einer Geraden:* Entfernen wir eine Gerade  $\theta_j$  aus dem momentanen Modell  $\theta$ , ohne damit die Wahrscheinlichkeit über alle Punkte im Wesentlichen zu verringern, deutet dies darauf hin, dass die Gerade  $\theta_j$  doppelt existiert bzw. dass der Geraden zu wenig Punkte zugeordnet werden. Um solch eine Gerade zu finden, wird ein Gütemaß des Modells  $\theta$  einmal mit und einmal ohne  $\theta_j$  berechnet und anschließend verglichen.

Nach der Konvergenz eines Modells zu einem lokalen Maximum werden diese zwei Sachverhalte überprüft und es wird entschieden, ob eine oder mehrere Geraden aus dem momentanen Modell entfernt werden und ob eine neue Gerade dem Modell hinzugefügt wird. In der Praxis hat sich herausgestellt, dass sowohl das Hinzufügen als auch das Entfernen einer Geraden in einem Iterationsschritt geschehen kann. Dies liegt daran, dass einer großen Menge von Messpunkten, die zu einer linearen Struktur gehören, zwei oder mehrere Geraden zugeordnet werden können. Diese Geraden liegen am Ende der Maximierung meistens übereinander, unabhängig davon ob es noch andere Punkte gibt, die nicht so gut durch eine Geraden erklärt werden. In diesem Fall muss sowohl eine Gerade entfernt als auch eine neue Gerade hinzugefügt werden. Deshalb müssen am Ende jeder Maximierung beide Sachverhalte überprüft werden.

### **Geraden entfernen**

Zuerst prüfen wir den Nutzen jeder Geraden und entfernen dann alle redundanten Geraden aus einem Modell. Dadurch wird die Komplexität eines Modells, das heißt die Anzahl der Geraden, beschränkt, da ein Modell nicht aus beliebig vielen Geraden bestehen soll. Wir führen zunächst das Gütemaß  $G^{[v]}$  für ein Geradenmodell  $\theta^{[v]}$  ein.

$$G^{[v]} = \sum_i d_{\min}(z_i, \theta^{[v]}) \quad (4.14)$$

wobei  $d_{\min}(z_i, \theta^{[v]})$  der minimale Abstand eines Punktes  $z_i$  zu einer Geraden im Modell  $\theta^{[v]}$  ist. Je kleiner der Wert  $G^{[v]}$  ist, um so besser wird die Menge der Messpunkte durch das Modell  $\theta^{[v]}$  beschrieben. Um die Anzahl der Geraden im Modell zu beschränken, führen wir einen sogenannten Strafterm  $M^{[v]} \cdot PENALTY$  ein, der zum Gütemaß addiert wird. Dabei ist  $M^{[v]}$  die Anzahl der Geraden im Modell  $\theta^{[v]}$  und  $PENALTY$  ein Parameter, der den Strafterm gewichtet. Die Kosten für eine einzelne Gerade sind folglich gleich dem Wert von  $PENALTY$ . Nun können wir den Nutzen jeder einzelnen Geraden im Modell berechnen und damit entscheiden, welche Geraden aus dem momentanen Modell zu entfernen sind. Der Nutzen einer einzelnen Geraden  $\theta_j^{[v]}$  im Modell  $\theta^{[v]}$  wird als  $N_j^{[v]}$  bezeichnet und wie folgt berechnet:

$$\begin{aligned} N_j^{[v]} &= [G^{[v-1]} + M^{[v-1]} \cdot PENALTY] - [G^{[v]} + M^{[v]} \cdot PENALTY] \\ &= [G^{[v-1]} - G^{[v]}] - PENALTY \end{aligned}$$

Dabei ist  $\theta^{[v-1]}$  das Modell  $\theta^{[v]}$  ohne die Gerade  $\theta_j^{[v]}$ . Ist nun der Wert  $N_j^{[v]} > 0$ , dann ist die Differenz der Gütemaße größer als  $PENALTY$ . Mit anderen Worten, die Gerade  $\theta_j^{[v]}$  verringert die Summe der minimalen Distanzen des Gütemaßes aus (4.14) um mehr als den Wert  $PENALTY$ . Ist der Nutzen einer Geraden größer 0, dann bleibt diese Gerade im Modell vorhanden, andernfalls kann diese Gerade aus dem Modell entfernt werden. Dadurch bleiben nur die Geraden in unserem Modell, welche das Gütemaß um mehr als den Wert von  $PENALTY$  verbessern und somit zur Erklärung des Punkte-modells wesentlich beitragen.

In der Praxis ist zu beachten, dass die Nutzen aller Geraden nacheinander berechnet werden müssen. Wird bei einer Geraden ein Nutzen kleiner 0 festgestellt, dann muss diese Gerade sofort entfernt werden, bevor die anderen Nutzen der Geraden berechnet werden. Ansonsten würden, wenn zwei Geraden sehr ähnlich sind, unter Umständen beide entfernt werden.

Der Parameter  $PENALTY$  ist sehr wichtig für unser Verfahren, da er mit für die Komplexität und Qualität des Modells verantwortlich ist. In *Abschnitt 5.2.2* untersuchen wir den Einfluss dieses Parameters auf die erstellten Modelle und versuchen für die Praxis geeignete Werte zu finden.

### Gerade hinzufügen

Nach dem Entfernen von redundanten Geraden müssen wir entscheiden, ob eine neue Gerade dem übrig gebliebenen Modell hinzugefügt wird. Das Einfügen einer neuen Geraden ist ein sehr wichtiger Schritt und kann nicht beliebig durchgeführt werden, da sonst zu häufig nach der Maximierung redundante Geraden entstehen.



Die Lage der neuen Geraden ist von großer Bedeutung, da die Gerade das bisherige Modell verbessern soll. Deshalb werden Punkte bestimmt, die schlecht durch das Geradenmodell erklärt werden, bzw. deren minimaler Abstand zu einer Geraden größer als  $NEW\_DIST\_MIN$  ist. Ist diese Menge von Punkten leer, dann kann das gesamte Verfahren abgebrochen werden, da alle Punkte gut genug durch das Modell erklärt werden. Ist die Menge jedoch nicht leer, fügen wir eine neue Gerade ein. Dafür wird aus dieser Menge ein Punkt zufällig bestimmt. Die neue einzufügende Gerade verläuft dann durch den zufällig gewählten Punkt, wobei die Ausrichtung der Geraden noch geschickt festgelegt werden muss.

Der zufällig gewählte Punkt gehört zu einer bestimmten linearen Struktur in einer Umgebung. Bezeichnen wir im Folgenden den zufällig gewählten Punkt mit  $p_0$ , die lineare Struktur des Punktes mit  $s_0$  und die neu eingefügte Gerade mit  $g_0$ . Da  $p_0$  nicht gut durch eine Gerade aus dem bisherigen Modell erklärt wird, wird vermutlich auch  $s_0$  nicht gut durch eine Gerade beschrieben. Somit gibt es wahrscheinlich noch weitere Punkte von  $s_0$ , die nicht gut durch das Modell beschrieben werden. Bei der Optimierung des Geradenmodells sollte sich  $g_0$  an  $s_0$  anpassen. Dies kann jedoch nicht immer garantiert werden, da wir mit einer Geraden arbeiten, die zumeist mehrere Hindernisse der Umgebung schneidet und somit nicht nur von den Punkten von  $s_0$  beeinflusst wird. Wir setzen deshalb die Ausrichtung der neuen Geraden  $g_0$  gleich der lokalen Ausrichtung des zufällig gewählten Punktes  $p_0$  und erhalten nach der Optimierung zumeist eine akzeptabel an  $s_0$  angepasste Gerade.

In zahlreichen Experimenten hat sich herausgestellt, dass bei einer zufälligen Wahl der Ausrichtung von  $g_0$  die eigentliche lineare Struktur  $s_0$  nach der Optimierung in vielen Fällen nicht beschrieben wurde (vgl. *Abschnitt 5.3.2*). Die Gerade  $g_0$  konnte sich nicht an  $s_0$  anpassen, da sie von zu vielen anderen Hindernissen beeinflusst wurde.

Die Ausrichtung von  $g_0$  ist bereits beim Einfügen sehr wichtig, damit  $s_0$  zu Beginn der Optimierung ungefähr beschrieben wird. Durch eine gut gewählte Ausrichtung wird  $g_0$  stark genug von den Punkten von  $s_0$  beeinflusst und wir können mit großer Wahrscheinlichkeit annehmen, dass  $g_0$  nicht zu einer redundanten Gerade konvergiert, sondern sich an  $s_0$  anpasst. In *Abschnitt 5.3.2* werden einige Experimente genauer analysiert, die diesen Sachverhalt und den Einfluss der lokalen Ausrichtung auf die erstellten Modelle etwas verdeutlichen.

Auch mit der lokalen Ausrichtung kann nicht immer garantiert werden, dass die neue Gerade nicht zu einem Duplikat konvergiert. Ein Duplikat entsteht vor allem dann, wenn der zufällig gewählte Messpunkt nicht von einem linearen Hindernis stammt. Die neue Gerade liegt daher wahrscheinlich nicht bei einer größeren Menge von schlecht beschriebenen Punkten und konvergiert deshalb etwas unkontrolliert. Solche zufällig gewählten Messpunkte werden meist nur schlecht durch ein Geradenmodell

beschrieben und können im Verlaufe des Verfahrens öfters zufällig ausgewählt werden. Diese Punkte sind ein großes Problem für das Verfahren und deshalb versuchen wir sie bereits zu Beginn aus dem Punktemodell zu filtern.

### 4.5 Abbruchkriterium

Die Unbekanntheit der optimalen Anzahl  $J$  von Geraden für ein Modell erschwert die Festlegung eines geeigneten Abbruchkriteriums. Die Anzahl der Geraden ist natürlich von der Umgebung des Roboters abhängig. Wie im Abschnitt zuvor erklärt, versuchen wir uns  $J$  im Verlauf der Optimierung anzunähern, um es somit bestmöglich zu bestimmen. Nachdem sich die Anzahl der Geraden verändert, wird erneut das wahrscheinlichste Modell mittels EM gesucht. Jedoch zu welchem Zeitpunkt können wir sagen, dass dem momentane Modell keine neue Gerade hinzugefügt oder keine Gerade entfernt werden muss? Mit anderen Worten: Zu welchem Zeitpunkt kann die Wahrscheinlichkeit des Punktemodells nicht mehr entscheidend verbessert werden?

Wenn wir im Verlauf des Verfahrens ein Modell erreichen, das alle Messungen gut erklärt, das heißt, dass der minimale Abstand jeder Messung zu einer Geraden kleiner als  $NEW\_DIST\_MIN$  ist, dann können wir dieses Geradenmodell als Endresultat verwenden. Jede Messung ist dann mit einer hohen Erwartung einer bestimmten Geraden zugeordnet. Der Vergleich der minimalen Abstände mit  $NEW\_DIST\_MIN$  ist somit ein gutes Abbruchkriterium während der Suche nach der optimalen Anzahl  $J$  von Geraden.

Diese Situation kann jedoch nicht immer garantiert erreicht werden, da es meistens eine Menge von Messungen gibt, die nur sehr schlecht einer Geraden zugeordnet werden können; ihre minimalen Abstände zu einer Geraden sind größer als  $NEW\_DIST\_MIN$ . Wir bezeichnen die gesamte Menge solcher Messungen mit  $Z_{error}$ . Bei der Vermessung einer Umgebung entstehen diese Messungen vor allem in folgenden Fällen:

- *Ausreißer*: Dies sind Messungen, die sehr weit von ihrer eigentlichen lineare Struktur entfernt sind. Da in ihrer nahen Umgebung nur wenig andere Messungen liegen, können Ausreißer nur schlecht zu Geraden zugeordnet werden.
- *nicht-lineare Strukturen*: Die Messungen nicht-linearer Strukturen können nur schwer durch Geraden erklärt werden. Solche Strukturen müssten durch mehrere Geraden beschrieben werden.
- *sehr kurze lineare Strukturen*: Bei sehr kurzen linearen Strukturen existieren nur wenig Messungen, die dieser Struktur angehören. Deshalb ist es wie bei Ausreißern schwierig, diesen Punkten eine Gerade zuzuordnen.

Solche Messungen aus  $Z_{error}$  versuchen wir zwar im Voraus herauszufiltern (*Abschnitt 2.1.3* und *2.2.3*), aber dies gelingt nur selten vollständig. Durch Einfügen einer neuen Gerade in das Modell könnten wir Messungen aus  $Z_{error}$  besser erklären, aber die neuen Geraden konvergieren anschließend entweder zu einem Duplikat oder ihr Nutzen ist sehr gering. Deshalb werden diese Geraden später wieder aus dem Modell entfernt und die Messungen bleiben weiterhin schlecht erklärt. Wenn wir feststellen könnten, dass nur noch Messungen aus  $Z_{error}$  einen minimalen Abstand zu einer Geraden größer  $NEW\_DIST\_MIN$  haben, dann könnten wir unser Verfahren anhalten. Daher erweitern wir unser bisheriges Abbruchkriterium.

Wie bereits beim Entfernen von Geraden in *Abschnitt 4.4* erläutert, sind wir nur an Geraden interessiert, die das Gütemaß des Modells wesentlich verbessern. Beim Einfügen einer neuen Geraden in ein momentanes Modell werden alle Punkte mit einem größeren minimalen Abstand zu einer Geraden als  $NEW\_DIST\_MIN$  bestimmt. In dieser Menge, bezeichnen wir sie mit  $Z_{new}$ , ist auch  $Z_{error}$  enthalten. Die Lage einer neuen Geraden wird anhand eines zufällig ausgewählten Punkt aus  $Z_{new}$  bestimmt. Ist der zufällig gewählte Punkt nicht in  $Z_{error}$  enthalten, dann können wir mit großer Wahrscheinlichkeit annehmen, dass die Gerade im Verlauf des Verfahrens einen Nutzen nach Konvergenz größer als 0 annimmt und somit nicht entfernt wird. Ist der Punkt jedoch in  $Z_{error}$  enthalten, dann wird die Gerade, wie bereits beschrieben, wahrscheinlich entfernt, da ihr Nutzen nach Konvergenz kleiner gleich 0 ist. In unserem Abbruchkriterium überprüfen wir nun, wie oft es hintereinander vorkommt, dass das Gütemaß  $G^{[v]}$  eines Modells  $\theta^{[v]}$  durch Einfügen einer Geraden nicht verbessert werden kann. Geschieht dies mehr als  $ERROR\_MIN$ -mal hintereinander, dann können wir annehmen, dass  $Z_{new}$  zum größten Teil nur noch aus Punkten der Menge  $Z_{error}$  besteht; mit einem geeigneten Wert für  $ERROR\_MIN$  können wir davon ausgehen, dass ein Geradenmodell durch Einfügen einer neuen Gerade nicht mehr verbessert werden kann.

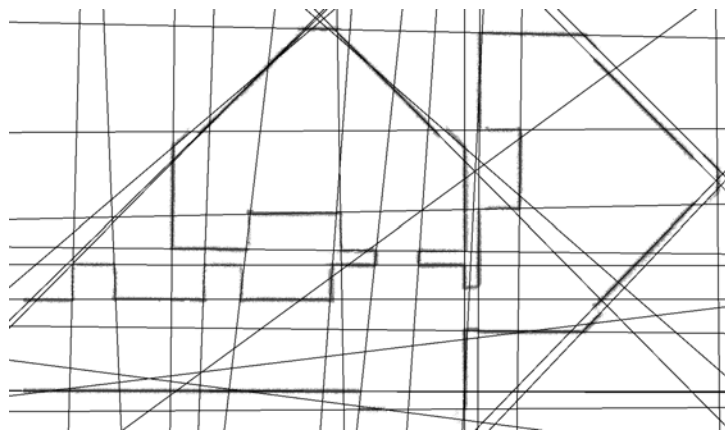
Das gesamte Abbruchkriterium wird durch  $NEW\_DIST\_MIN$  und  $ERROR\_MIN$  festgelegt. Für die Praxis geeignete Werte lassen sich für  $NEW\_DIST\_MIN$  relativ einfach bestimmen. Damit ein Messpunkt ausreichend gut von einer Geraden beschrieben wird, sollte er maximal 10 cm von einer Geraden entfernt liegen. Punkte mit sehr geringem minimalen Abstand werden durch eine Gerade gut beschrieben und sollen bei der zufälligen Wahl eines Punktes nicht berücksichtigt werden. Empfehlenswert sind daher Werte zwischen 5 cm und 10 cm für den Parameter  $NEW\_DIST\_MIN$ . Die Belegung des anderen Parameters  $ERROR\_MIN$  kann nicht so einfach abgeschätzt werden, da ein geeigneter Wert auch von der Größe von  $Z_{error}$  abhängt. Je größer diese Menge ist, um so größer müssen wir auch den Wert von  $ERROR\_MIN$  belegen, damit wir nicht zu früh abbrechen. Ein Geradenmodell kann ein Punktemodell nur dann gut beschreiben, wenn die optimale Anzahl von Geraden für das Modell annähernd erreicht wird. In zahlreichen von uns durchgeführten Versuchen hat sich  $ERROR\_MIN = 15$  als

gut erwiesen und man konnte aus den meisten Punktemodellen ein akzeptables Geradenmodell mit einer annähernd optimalen Anzahl von Geraden bilden. Nur bei Punktemodellen mit vielen fehlerhaften Messungen mussten wir den Wert von *ERROR\_MIN* erhöhen, damit das Verfahren nicht zu früh abbrach. Theoretisch könnten wir *ERROR\_MIN* beliebig weit erhöhen, ohne das Ergebnis großartig zu verändern, aber dadurch würde das Verfahren unnötig länger laufen.

### 4.6 Übergang von Geraden zu Liniensegmenten

Bei der Erwartungsmaximierung haben wir mit Geraden gearbeitet, die unsere Messungen bestmöglich erklären sollten. Die Verwendung von Geraden hat uns das bisherige Vorgehen sehr vereinfacht, aber die Geradenmodelle beschreiben die Umgebungen nicht ausreichend gut, wie man in einem Beispiel in *Abbildung 4.1* sehen kann. Ein Geradenmodell muss nun in ein Linienmodell umgewandelt werden, das aus Liniensegmenten besteht.

Der Übergang zu Liniensegmenten bei diesem EM-Ansatz unterscheidet sich ein wenig von der Segmentierung der Geraden des heuristischen Verfahrens aus *Abschnitt 3.2*. Nach der Bestimmung einer Geraden wird diese dort sofort segmentiert, bevor eine weitere Gerade anhand der verbleibenden Punkte bestimmt wird. Diese schrittweise Bestimmung von Geraden mit anschließender Segmentierung können wir in unserer jetzigen Situation nicht mehr anwenden, da wir nun eine Menge von Geraden besitzen, deren Elemente sozusagen gleichzeitig segmentiert werden müssen.



**Abbildung 4.1:** Mit dem EM-Verfahren erstelltes Geradenmodell (39 Geraden) und im Hintergrund das zugrunde liegende Punktemodell (133.134 Punkte)

Wir wollen die Segmentierung eines optimalen Geradenmodells  $\theta^{opt}$ , das aus  $J$  Geraden  $\theta_j^{opt}$  besteht, durchführen. Dafür verwenden wir zum einen alle Messungen  $z_i \in Z$  und zum anderen alle Erwartungen  $E[c_{ij}|\theta^{opt}, Z]$ . Ähnlich der Segmentierung im heuristischen Verfahren benötigen wir auch hier drei wesentliche Schritte, die zum Teil analog übernommen werden können:

### Punkte auf die Geraden verteilen

Um eine einzelne Gerade  $\theta_j^{opt}$  zu segmentieren, müssen wir entscheiden, welche Messungen wir dafür berücksichtigen. Wir können nicht wie beim heuristischen Verfahren alle Messungen berücksichtigen, die nah genug an der Geraden liegen, da mehrere Geraden sozusagen gleichzeitig segmentiert werden müssen, wodurch ein Punkt mehreren Geraden zugeordnet werden könnte. Deshalb verwenden wir die Erwartungen  $E[c_{ij}|\theta^{opt}, Z]$ , um dies zu vermeiden. Für die Segmentierung einer Geraden  $\theta_j^{opt}$  werden nur die Messungen  $z_i \in Z$  berücksichtigt, deren Erwartung, dieser Geraden zugeordnet zu werden, am größten ist. Wir bezeichnen diese Menge von Messungen als  $Z_j^{opt}$ . Mathematisch bedeutet dies für alle Punkte  $z_i \in Z_j^{opt}$  folgendes:

$$E[c_{ij}|\theta^{opt}, Z] > E[c_{ik}|\theta^{opt}, Z]; \quad \forall 1 \leq k \leq J; \quad j \neq k$$

Somit wird jede Messung nur bei einer Segmentierung einer Geraden berücksichtigt. Das heißt nichts anderes, als dass jede Messung nur einer Geraden zugeordnet wird. Mit diesem Sachverhalt kann nun jeder Geraden  $\theta_j^{opt}$  die Menge  $Z_j^{opt}$  eindeutig zugeordnet werden und es ist somit egal, in welcher Reihenfolge wir die einzelnen Geraden segmentieren.

Im E-Schritt wurde bereits erklärt, wie die Erwartungen  $E[c_{ij}|\theta^{opt}, Z]$  berechnet werden (*Abschnitt 4.3.1*). Dabei hängt eine einzelne Erwartung nur von den Abständen eines Punktes  $z_i$  zu den Geraden aus  $\theta^{opt}$  ab und berücksichtigt somit nur die globalen Beziehungen zwischen einem Punkt und den Geraden. Deshalb haben wir eine erweiterte Erwartung  $E^*[c_{ij}|\theta^{opt}, Z]$  eingeführt, die zusätzlich von der lokalen Ausrichtung eines Punktes  $z_i$  und den Ausrichtungen aller Geraden aus  $\theta^{opt}$  abhängt. Vor allem bei der Segmentierung ist diese erweiterte Erwartung sehr vorteilhaft. Liegt ein Punkt ungefähr gleich weit von zwei oder mehreren Geraden entfernt, dann wird er nicht unbedingt der Geraden mit dem kürzesten Abstand zugeordnet, sondern der Geraden, deren Ausrichtung am wenigsten von der lokalen Ausrichtung des Punktes abweicht. Bei der Verwendung von  $E^*[c_{ij}|\theta^{opt}, Z]$  wird somit nicht nur die globale Beziehung zwischen den Messungen und den Geraden berücksichtigt, sondern auch die während des Scanvorgangs erhaltenen Informationen der lokalen Ausrichtung. Der Einfluss auf die praktischen Ergebnisse wird im *Abschnitt 5.3.2* näher erläutert.

### **Einzelne Geraden segmentieren**

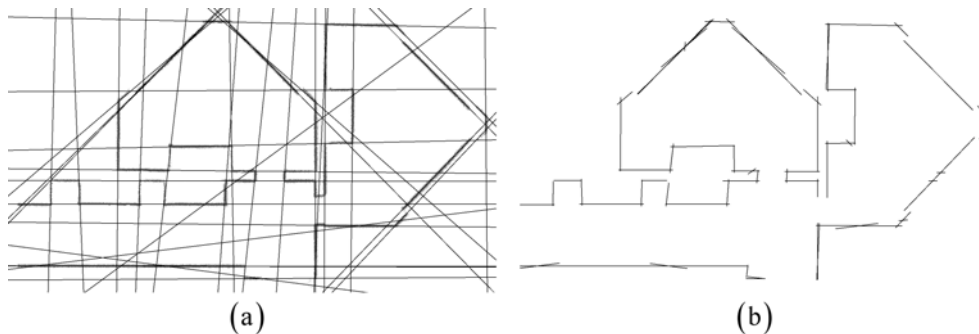
Nachdem für jede Gerade  $\theta_j^{opt}$  die Menge  $Z_j^{opt}$  bestimmt wurde, kann die eigentliche Segmentierung der Geraden beginnen. Da wir die Geraden getrennt voneinander segmentieren können, kann der Ansatz des heuristischen Verfahrens aus *Abschnitt 3.2.2* analog übernommen werden. In diesem Ansatz wird der Parameter *SEG\_DIST\_MAX* benötigt, der eine korrekte und für das Modell sinnvolle Segmentierung gewährleistet.

### **Filterung der Liniensegmente einer Geraden**

Wurde eine Gerade in mehrere Liniensegmente unterteilt, müssen nur noch fehlerhafte Segmente entfernt werden, die nicht für unser späteres Linienmodell geeignet sind. Ein einzelnes Liniensegment sollte im optimalen Fall eine komplette lineare Struktur der Umgebung beschreiben. Da wir zu Beginn dieses EM-Verfahrens mit Geraden arbeiten, kann eine Gerade eine lineare Struktur schneiden und diese nicht komplett überdecken. In solch einem Fall entsteht ein Liniensegment, das als fehlerhaft angesehen werden kann, da dieses Segment nicht die komplette lineare Struktur beschreibt und falsch ausgerichtet ist.

Ähnlich dem heuristischen Verfahren kann eine Filterung fehlerhafter Liniensegmente durchgeführt werden, wobei der größte Unterschied zu *Abschnitt 3.2.3* darin liegt, dass nicht nur ein Liniensegment einer Geraden übrig bleiben darf, sondern auch mehrere. Wir übernehmen aus *Abschnitt 3.2.3* den Parameter *SEG\_LENGTH\_MIN*, der die Mindestlänge eines zulässigen Liniensegments definiert. Besitzt ein Liniensegment eine geringere Länge als *SEG\_LENGTH\_MIN*, dann wird das Segment als zu kurz, bzw. als fehlerhaft angenommen und es wird gelöscht. In der jetzigen Situation ist dieser Parameter sehr wichtig, da eine Gerade nicht nur eine lineare Struktur überdeckt, sondern auch zumeist mehrere Hindernisse schneidet und dadurch einige fehlerhafte Liniensegmente entstehen können. Am Ende der Filterung bleiben nur die Liniensegmente mit einer größeren Länge als *SEG\_LENGTH\_MIN* übrig und die meisten fehlerhaften Segmente können vermieden werden.

Mit diesen drei Schritten kann die Segmentierung einer Geraden durchgeführt werden. Wird jede Gerade aus dem Geradenmodell segmentiert und anschließend alle entstandenen Segmente zu einem Linienmodell zusammengefasst, dann kann der Übergang von einem Geradenmodell hin zu einem Linienmodell als abgeschlossen angesehen werden. In *Abbildung 4.2* ist eine Segmentierung eines Geradenmodells dargestellt, wobei die zwei Parameter *SEG\_LENGTH\_MIN* = 20 cm und *SEG\_DIST\_MAX* = 5 cm verwendet wurden. Für diese Parameter eignen sich in der Praxis Werte, die identisch mit den Werten aus dem heuristische Verfahren sind (vgl. *Abschnitt 5.2.1*).



**Abbildung 4.2:** Ausgehend vom Geradenmodell mit 39 Geraden in (a) ist mit der Segmentierung das Liniensegmentmodell mit 62 Liniensegmenten in (b) erstellt worden, wobei die Parameter  $SEG\_DIST\_MAX = 5$  cm und  $SEG\_LENGTH\_MIN = 20$  cm verwendet wurden.

## 4.7 Bearbeitung der Liniensegmente

Die in diesem Kapitel erstellten Liniensegmente werden durch verschiedene Ansätze verbessert. Wird jedes entstandene Liniensegment mit den anderen eines Modells anhand der Lage verglichen, dann können einige Liniensegmente gelöscht und die meisten anderen Liniensegmente verändert werden. Dadurch erhalten wir eine geringere Komplexität und ein besseres Liniensegmentmodell als zuvor.

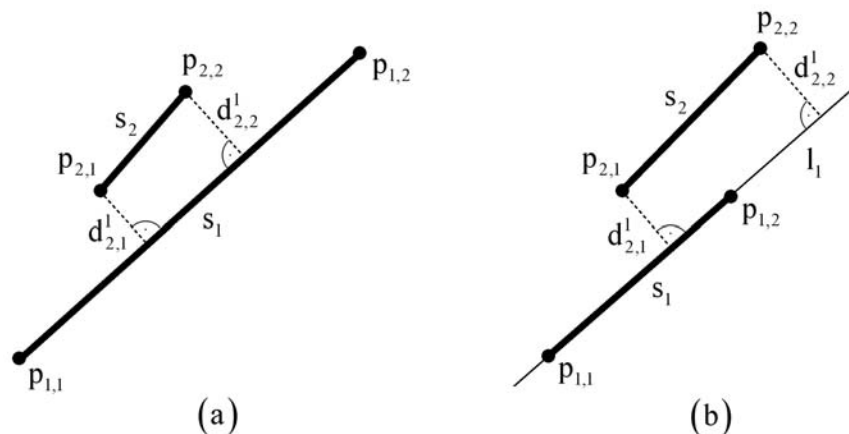
Solch eine Verbesserung der Modelle wird auch im heuristischen Verfahren in *Abschnitt 3.5* vorgestellt. Da wir unsere jetzigen Liniensegmentmodelle etwas anders erstellt haben, können wir die Ideen aus *Abschnitt 3.5* nur teilweise übernehmen. Die Unterschiede und Parallelen werden in den nächsten Abschnitten erklärt.

### 4.7.1 Löschen von Liniensegmenten

Bei der Segmentierung einer Geraden kann zum Beispiel ein Liniensegment entstehen, welches eine lineare Struktur nur sehr schlecht beschreibt. Dieses Segment kann gelöscht werden, wenn ein anderes Liniensegment existiert, welches die lineare Struktur besser bzw. komplett beschreibt (siehe *Abbildung 4.2 (b)*). Im heuristischen Verfahren werden solche Liniensegmente vereinigt, in unserem jetzigen Verfahren hingegen ist das Löschen eines der beiden Liniensegmente vorteilhafter.

Für das Löschen von Liniensegmenten übernehmen wir die Definition des Abstandes  $d_i^j$  und der Überlappung  $o_i^j$  eines Endpunktes  $p_i$  mit einem anderen Liniensegment  $s_j$  aus *Abschnitt 3.5.1*. Um zu entscheiden, ob ein Liniensegment in einem Liniensegmentmodell überflüssig ist und gelöscht werden kann, vergleichen wir jedes Liniensegment mit allen anderen. Angenommen wir vergleichen das Segment  $s_j$  mit

$s_j$  und prüfen, ob  $s_2$  gelöscht werden kann, dann werden zuerst die Abstände  $d_{2,1}^l$  und  $d_{2,2}^l$  der beiden Endpunkte  $p_{2,1}$  und  $p_{2,2}$  zum Segment  $s_j$  berechnet. Ist nur einer dieser Abstände größer als  $DELETE\_DIST\_MAX$ , dann ist  $s_2$  zu weit von  $s_j$  entfernt und ein Löschen von  $s_2$  wäre nicht sinnvoll. Sind jedoch beide Abstände kleiner als  $DELETE\_DIST\_MAX$ , dann überprüfen wir, ob  $s_2$  mit  $s_j$  komplett überlappt. Überlappen beide Endpunkte  $p_{2,1}$  und  $p_{2,2}$  mit dem anderen Segment  $s_j$ , dann sind  $o_{2,1}^l = 1$  und  $o_{2,2}^l = 1$ , nach Definition in *Abschnitt 3.5.1*. Das heißt nichts anderes, als dass  $s_2$  komplett mit  $s_j$  überlappt und somit kann  $s_2$  gelöscht werden. Ist nur ein Wert von  $o_{2,1}^l$  und  $o_{2,2}^l$  gleich 1, dann darf  $s_2$  nicht gelöscht werden. Zur Verdeutlichung sind zwei Situationen mit und ohne kompletter Überlappung in *Abbildung 4.3* dargestellt.

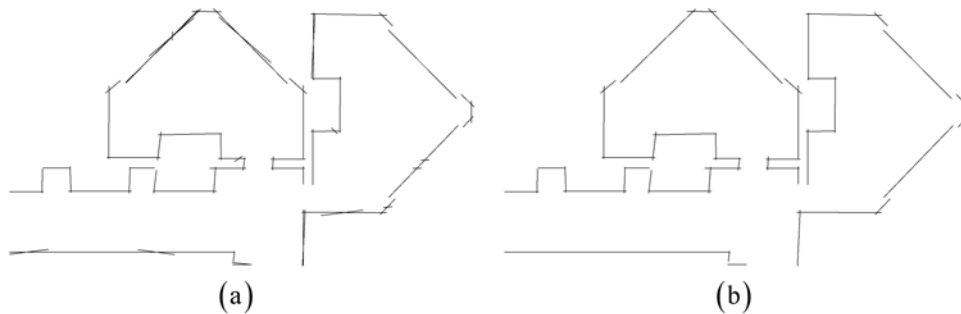


**Abbildung 4.3:** Beide Skizzen stellen zwei Liniensegmente dar, mit in (a) kompletter und in (b) teilweiser Überlappung.  $s_2$  kann in (a) in Abhängigkeit von den Abständen  $d_{2,1}^l$  und  $d_{2,2}^l$  unter Umständen, und in (b) auf keinen Fall gelöscht werden.

Warum wir diese zwei Liniensegmente  $s_j$  und  $s_2$  nicht vereinigen liegt daran, dass die Lage der Geraden  $l_j$ , aus der  $s_j$  entsteht, bereits bei ihrer Berechnung durch die Messpunkte von  $s_2$  beeinflusst wurde. Eine Vereinigung, die unsere EM-Idee nicht verfälscht, ist nicht so einfach durchzuführen. Für eine korrekte Gewichtung der Segmente bzw. der Messpunkte müsste eigentlich die Gerade  $l_2$  aus unserem Geradenmodell entfernt und der EM-Ansatz nochmals durchgeführt werden, damit wir korrekte Erwartungen für die Messpunkte erhalten. Dies würde aber unser ganzes Modell verändern und unter Umständen sogar verschlechtern. Deshalb wenden wir keine Vereinigung an, sondern löschen die überflüssigen Liniensegmente.

Das Löschen von Liniensegmenten in der Praxis wird durch ein Beispiel in *Abbildung 4.4* dargestellt. Die Wahl eines geeigneten Wertes für  $DELETE\_DIST\_MAX$  ist vergleichbar mit der Wahl des Wertes für den Parameter  $MERGE\_DIST\_MAX$  aus dem heuristischen Verfahren. In *Abbildung 4.4* wurde deshalb der Wert von  $DELETE\_DIST\_MAX$  auf 10 cm gesetzt.



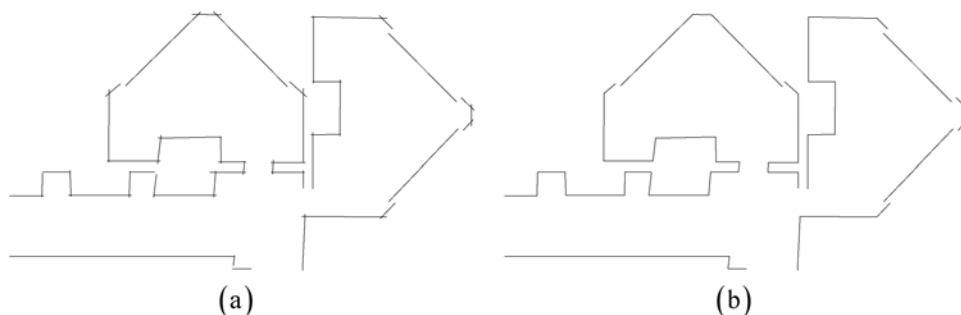


**Abbildung 4.4:** Das Linienmodell mit 62 Liniensegmenten in (a) wurde durch Löschen von Segmenten in ein Linienmodell mit 46 Liniensegmenten in (b) umgewandelt, wobei der Parameter  $DELETE\_DIST\_MAX = 10$  cm verwendet wurde.

### 4.7.2 Veränderung der Länge der Liniensegmente

Wie bereits in *Abschnitt 3.5.2* des heuristischen Verfahrens erläutert, sollten die einzelnen Liniensegmente eines Linienmodells zusammenhängen. Für ein einzelnes Liniensegment bedeutet dies, dass dessen Endpunkte auf einem anderen Liniensegment liegen sollten. Nachdem wir mittels EM und dem Löschen von Liniensegmenten ein Linienmodell erstellt haben, sind die Liniensegmente jedoch nicht zusammenhängend. Um diesen Fehler in den Modellen zu beheben, verwenden wir analog das Verfahren zur Veränderung der Länge von Liniensegmenten aus *Abschnitt 3.5.2*. Wir können ohne Probleme dieses Verfahren übernehmen, da nur anhand der Lage der Liniensegmente entschieden wird, welche Liniensegmente verändert werden sollen. Es wird nicht die Ausrichtung, sondern nur die Länge eines Liniensegments verändert, wobei die Länge der Veränderung durch  $EXTEND\_LENGTH$  nach oben beschränkt ist. Somit bleibt die mittels EM erhaltene Ausrichtung der Segmente erhalten.

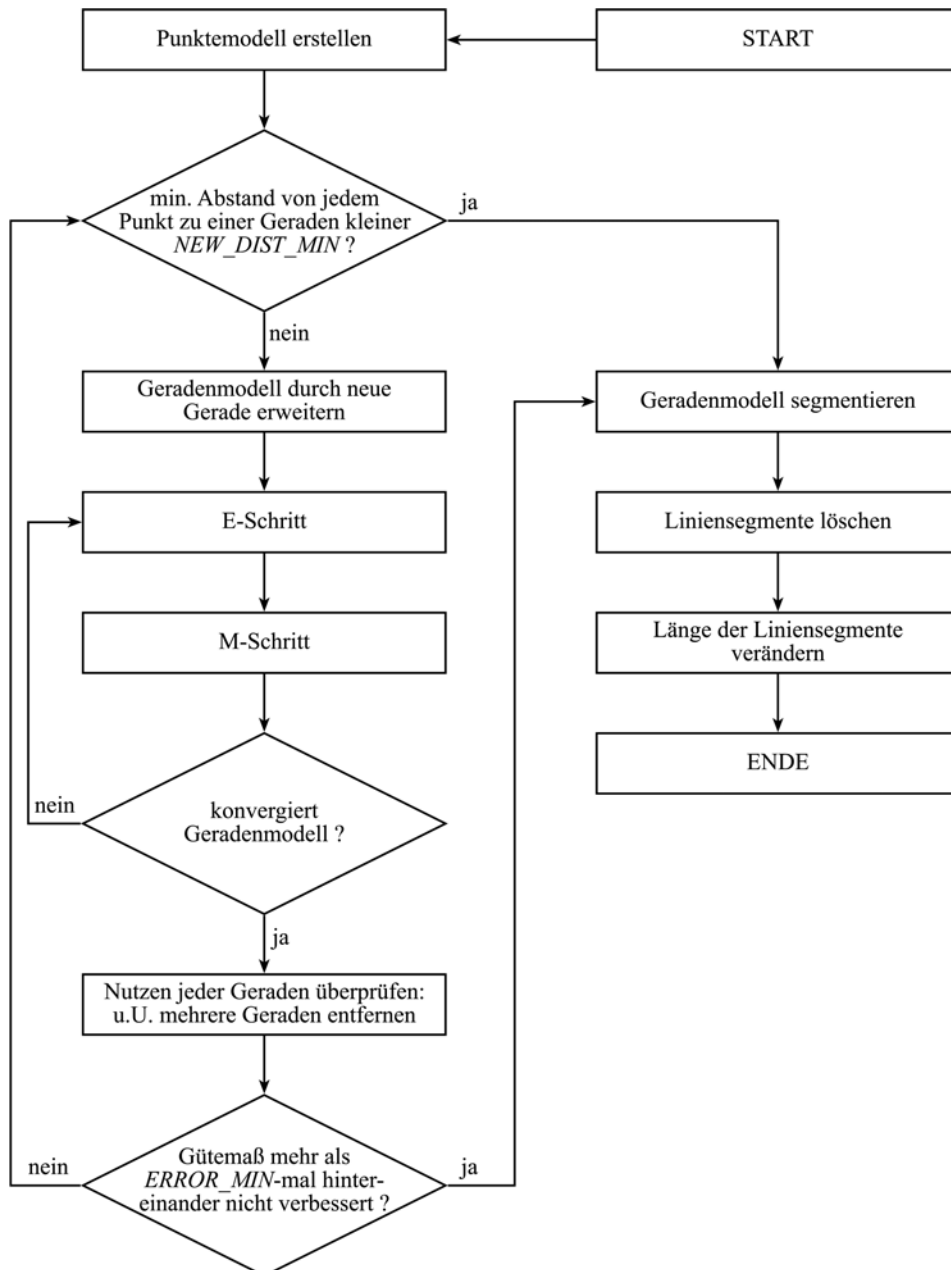
Die *Abbildung 4.5* zeigt ein Beispiel einer Veränderung der Länge der Liniensegmente eines Linienmodells. Für die Praxis eignen sich analog zum heuristischen Verfahren Werte für  $EXTEND\_LENGTH$  zwischen 15 cm und 25 cm.



**Abbildung 4.5:** (b) zeigt ein Linienmodell nach der Veränderung der Länge der Liniensegmente des Modells in (a), wobei der Parameter  $EXTEND\_LENGTH = 20$  cm verwendet wurde.

## 4.8 Flussdiagramm des Verfahrens

Das gesamte EM-Verfahren zur Erstellung von Linienmodellen wird durch ein Flussdiagramm in *Abbildung 4.6* visualisiert, um einen zusammenfassenden Überblick zu geben.



**Abbildung 4.6:** Das Flussdiagramm zeigt den Ablauf der wichtigsten Schritte des gesamten EM-Verfahrens zur Erstellung von Linienmodellen.

# 5 Experimente

In den Kapiteln zuvor haben wir zwei Verfahren vorgestellt, welche aus einem Punktemodell ein Linienmodell einer Umgebung des Roboters erstellen. Damit wir nun die Ergebnisse dieser Verfahren untersuchen und vergleichen können, stellen wir in diesem Kapitel als Erstes unterschiedliche Beurteilungskriterien für die Modelle vor. In den folgenden Abschnitten untersuchen wir die Ergebnisse von verschiedenen räumlichen Umgebungen der beiden Verfahren. In Abschnitt 5.2 behandeln wir den Einfluss und die Festlegung der Werte der wichtigsten Parameter. Danach betrachten wir den Einfluss der lokalen Ausrichtungen an den verschiedenen Stellen der Verfahren und beurteilen die erstellten Linienmodelle mit und ohne Berücksichtigung der lokalen Ausrichtung. Das Kapitel endet mit dem Vergleich der beiden vorgestellten Verfahren.

Für diese Analyse haben wir beide Verfahren in der Programmiersprache C++ implementiert. Die Daten der Vermessungen von verschiedenen Umgebungen wurden uns von Mitarbeitern der Arbeitsgruppe<sup>6</sup> zur Verfügung gestellt.

## 5.1 Beurteilungskriterien

Für die Beurteilung der erstellten Modelle führen wir drei Kriterien ein, die in den folgenden Abschnitten verwendet werden:

- 1.) Ein Kriterium, um verschiedene Linienmodelle zu beurteilen bzw. zu vergleichen, ist die Anzahl der erstellten Liniensegmente. Es entspricht dem grundsätzlichen Ziel der Verfahren, Modelle mit einer möglichst geringen Komplexität zu erstellen, natürlich unter Berücksichtigung der Lage der Liniensegmente und der vollständigen Beschreibung der Umgebung.
- 2.) Für das EM-Verfahren bilden wir wie in *Abschnitt 4.4* ein Gütemaß  $G(\theta, Z)$  für ein Geradenmodell  $\theta$ , gegeben der zugehörigen Punktemenge  $Z = \{z_i | i = 1, \dots, n\}$ :

$$G(\theta, Z) = \sum_{i=1}^n d_{\min}(z_i, \theta)$$

---

<sup>6</sup> Arbeitsgruppe Autonome Intelligente Systeme am Institut für Informatik, Albert-Ludwigs-Universität Freiburg

wobei  $d_{\min}(z_i, \theta)$  der minimale Abstand eines Messpunktes  $z_i$  zu einer Geraden aus  $\theta$  ist. Bei der Verwendung dieses Gütemaßes ist zu beachten, dass das Punktemodell nicht exakt der realen Welt entspricht, sondern auch nur ein Modell der Umgebung darstellt. Deshalb ist  $G(\theta, Z)$  ein Gütemaß für den Übergang vom Punktemodell  $Z$  hin zum Geradenmodell  $\theta$  und nicht ein Gütemaß des Geradenmodells für die Beschreibung der Umgebung des Roboters. Wir verwenden dieses Gütemaß nur zum Vergleichen verschiedener Geradenmodelle, die anhand der gleichen Punktemenge  $Z$  erstellt wurden.

3.) Bei der Erstellung von Liniensegmenten kann es oftmals zu einem sogenannten *overfitting* kommen, bei dem die Werte der Beurteilungskriterien gut sind, aber das Linienmodell die Umgebung nicht sehr gut repräsentiert. Um verschiedene Linienmodelle miteinander zu vergleichen, reicht ein Kriterium allein nicht aus. Wir verwenden daher in den folgenden Abschnitten vor allem eine visuelle Beurteilung der Linienmodelle. Bei einer genauen visuellen Betrachtung eines Linienmodells können dann kleine Fehler in den Liniensegmenten und überflüssige Liniensegmente gefunden werden, welche auf die Werte der Beurteilungskriterien zum Teil keine verschlechternde Wirkung haben.

Es ist nicht einfach, geeignete Beurteilungskriterien zu finden, da die einzigen Informationen über die Umgebung die Messpunkte sind. In unserer Situation ist die visuelle Beurteilung eines Linienmodells am Wichtigsten. Ein Linienmodell kann auch ohne ein anderes Modell allein visuell analysiert werden. Wir verwenden deshalb in den nächsten Abschnitten bei der Analyse der Linienmodelle hauptsächlich den visuellen Charakter der Modelle. Die anderen zwei vorgestellten Beurteilungskriterien werden zusätzlich zur visuellen Beurteilung untersucht und geben uns weitere Informationen über die Qualität der erstellten Linienmodelle.

## 5.2 Einfluss der Parameter

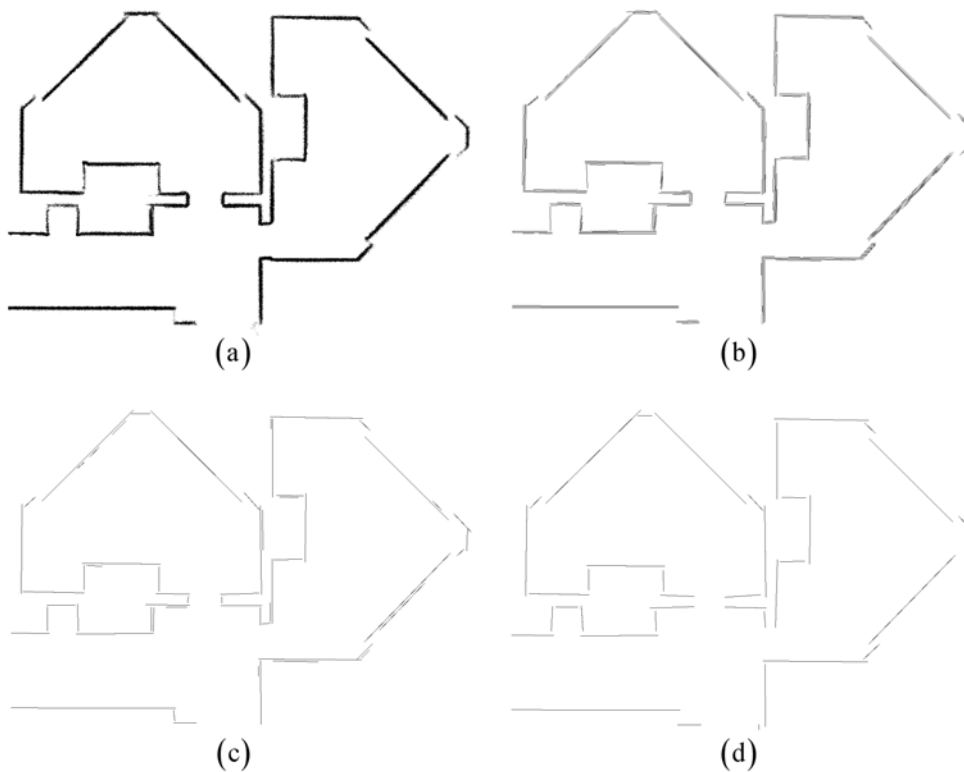
Bevor wir ein Punktemodell in ein Linienmodell umwandeln, müssen die Parameter eines Verfahrens mit geeigneten Werten belegt werden. Wir werden nun den Einfluss der wichtigsten Parameter auf die erstellten Linienmodelle erläutern und für diese Parameter geeignete Werte für die Praxis angeben.

### 5.2.1 Heuristisches Verfahren

In diesem Abschnitt werden die Parameter des heuristischen Verfahrens aus Kapitel 3 behandelt:

***ASSIGN\_DIST\_MAX***

Dieser Parameter legt den maximalen Abstand fest, den ein Messpunkt zu einer Geraden besitzen darf, damit der Punkt bei der Segmentierung der Geraden zugeordnet wird. *ASSIGN\_DIST\_MAX* legt somit fest, welche Punkte einer Geraden zugeordnet und bei der anschließenden Segmentierung berücksichtigt werden. Betrachten wir zunächst *Abbildung 5.1*, in der wir ein Punktemodell mit verschiedenen Werten für *ASSIGN\_DIST\_MAX* in Linienmodelle umgewandelt haben.



**Abbildung 5.1:** Ausgehend vom Punktemodell (a) mit 115.988 Messpunkten wurden Linienmodelle gebildet, mit Werten für *ASSIGN\_DIST\_MAX* von (b) 2 cm, (c) 5 cm und (d) 10 cm.

<b>Abbildung 5.1</b>	<b><i>ASSIGN_DIST_MAX</i></b>	<b>Liniensegmente</b>
(b)	2 cm	175
(c)	5 cm	62
(d)	10 cm	44

**Tabelle 5.1:** Anzahl der Liniensegmente der Linienmodelle aus *Abbildung 5.1*

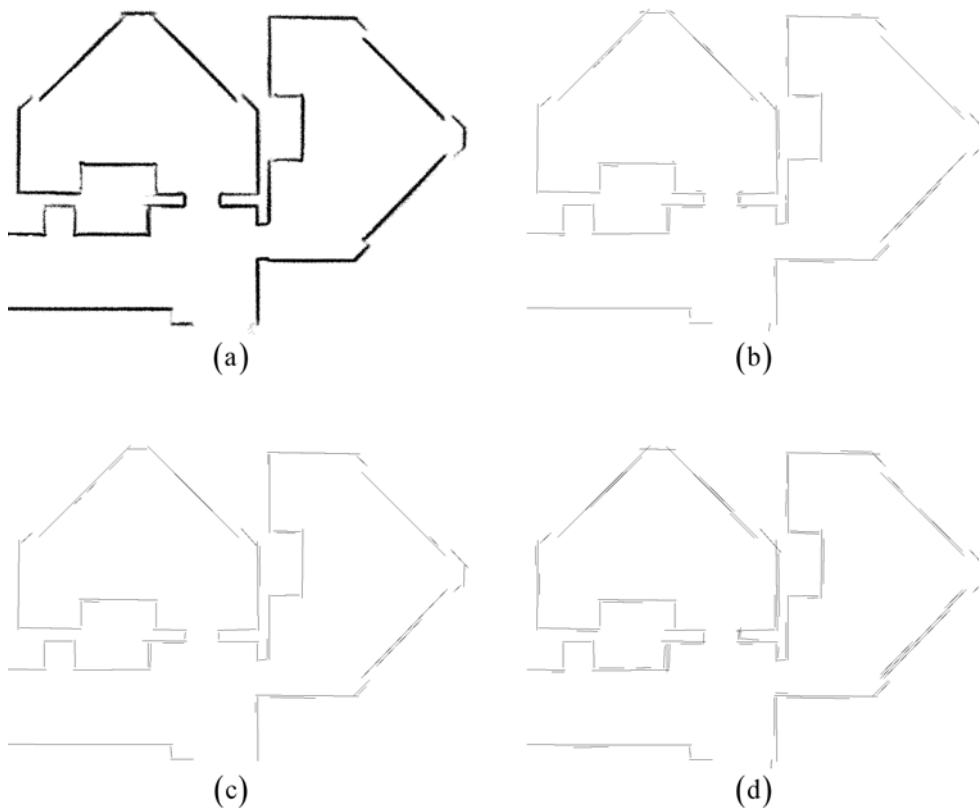
In Verbindung mit der *Tabelle 5.1* ist leicht zu erkennen, dass der Wert dieses Parameters für die Qualität und Quantität der Liniensegmente mit verantwortlich ist. Bei einem Wert von 2 cm wie in *Abbildung 5.1 (b)* wird eine einzelne lineare Struktur durch mehrere Liniensegmente beschrieben. Die erstellten Segmente sind zwar alle richtig ausgerichtet, eine lineare Struktur sollte aber eigentlich nur durch ein Liniensegment beschrieben werden. Legen wir den Wert auf 10 cm wie in *Abbildung 5.1 (d)*, dann wird eine lineare Struktur durch genau ein Liniensegment beschrieben. Das Problem bei diesem Wert ist, dass nicht jede lineare Struktur durch ein Segment beschrieben wird. Für die etwas kürzeren linearen Strukturen wurden keine Segmente erstellt, da deren Messpunkte anderen angrenzenden Hindernissen zugeordnet wurden. Durch diese fehlerhafte Zuordnung entsteht außerdem eine nicht optimale Lage der erstellten Liniensegmente. Bei genauer Betrachtung von *(d)* ist zu erkennen, dass die Segmente im Vergleich zu den linearen Strukturen teilweise leicht schräg liegen. Ein guter Kompromiss zwischen Quantität und Qualität der Liniensegmente bildet  $ASSIGN\_DIST\_MAX = 5$  cm, wie in *Abbildung 5.1 (c)*. Mit diesem Wert wird fast jede lineare Struktur durch mindestens ein Liniensegment beschrieben. Die Ausrichtung der Segmente ist bezüglich der Lage der Strukturen sehr gut. Einige Strukturen werden jedoch durch mehrere Segmente beschrieben. Mittels der Vereinigung von Liniensegmenten aus *Abschnitt 3.5.1* können wir die Anzahl der Segmente verringern, damit keine lineare Struktur durch mehrere Segmente beschrieben wird.

In den Versuchen mit verschiedenen Punktemodellen hat sich herausgestellt, dass die optimale Wahl des Wertes von  $ASSIGN\_DIST\_MAX$  auch von der Qualität der Messpunkte abhängig ist. Weichen die Messpunkte nur gering von ihrer linearen Struktur ab, dann kann  $ASSIGN\_DIST\_MAX$  kleiner gewählt werden als wenn der Fehler in den Messpunkten größer ist. In der Praxis haben sich Werte zwischen 5 cm und 10 cm für die meisten Punktemodelle als gut erwiesen, womit wir insgesamt einen Bereich mit einer Breite zwischen 10 cm und 20 cm erhalten in dem die Punkte liegen dürfen, damit sie einer Geraden zugeordnet werden.

### ***SEG\_DIST\_MAX* und *SEG\_LENGTH\_MIN***

Diese beiden Parameter werden bei der Segmentierung einer Geraden benötigt und im Folgenden gemeinsam behandelt. Um geeignete Werte für beide Parameter zu finden, betrachten wir *Abbildung 5.2* und *Tabelle 5.2*.

Die Festlegung des Parameters  $SEG\_DIST\_MAX$  ist von der Dichte der Lage der Messpunkte im Punktemodell abhängig. Werden bei der Exploration einer Umgebung die Scans schnell hintereinander durchgeführt, so liegen die einzelnen Messpunkte der linearen Strukturen dicht beieinander. Dies ist natürlich auch von der Bewegungsgeschwindigkeit des Roboters abhängig. Sind die Abstände zwischen den einzelnen sortierten Punkten einer linearen Struktur bei der Segmentierung gering, dann lässt



**Abbildung 5.2:** Ausgehend vom Punktemodell (a) mit 115.988 Messpunkten wurden Linienmodelle gebildet, mit Werten für  $SEG\_DIST\_MAX$  von (b), (c) 3 cm und (d) 10 cm, sowie für  $SEG\_LENGTH\_MIN$  von (b) 5 cm und (c), (d) 20 cm.

<b>Abbildung 5.2</b>	<b><math>SEG\_DIST\_MAX</math></b>	<b><math>SEG\_LENGTH\_MIN</math></b>	<b>Liniensegmente</b>
(b)	3 cm	5 cm	83
(c)	3 cm	20 cm	62
(d)	10 cm	20 cm	86

**Tabelle 5.2:** Anzahl der Liniensegmente der Linienmodelle aus *Abbildung 5.2*

sich diese Strukturen besser bestimmen. In *Abbildung 5.2 (a)* besitzt das Punktemodell eine hohe Dichte der Messpunkte. Vergleichen wir die beiden Linienmodelle in *Abbildung 5.2 (c)* und *(d)*, dann entstehen bei *(d)* wesentlich mehr überflüssige Liniensegmente. Der Grund dafür sind die Ausreißer in den Messpunkten einer linearen Struktur. Diese können meistens nicht mit den anderen Messpunkten der Struktur gemeinsam einer Geraden zugeordnet werden. Im Verlauf des Verfahrens können dann aus übrig gebliebenen Ausreißern Segmente gebildet werden. Da die Dichte der

Ausreißer gering ist, können wir bei der Segmentierung der Geraden durch einen niedrigen Wert für  $SEG\_DIST\_MAX$  die meisten überflüssige Liniensegmente vermeiden, wie in (c) dargestellt. Bei einer dichten Lage der Messpunkte eignen sich Werte für  $SEG\_DIST\_MAX$  zwischen 2 cm und 5 cm.

Aus dem gleichen Grund entstehen in *Abbildung 5.2 (b)* einige überflüssige Liniensegmente, die wir jedoch nicht durch ein kleines  $SEG\_DIST\_MAX$  vermeiden können. Wie in (b) zu erkennen ist, sind diese Segmente zumeist sehr kurz und werden in (c) durch einen größeren Wert von  $SEG\_LENGTH\_MIN$  vermieden. Durch die Verwendung dieses Parameters können alle linearen Strukturen mit einer kürzeren Länge als  $SEG\_LENGTH\_MIN$  nicht mehr durch ein Liniensegment beschrieben werden. Deshalb sollte dieser Wert nicht zu groß gewählt werden. Mit einem Wert von ca. 20 cm für  $SEG\_LENGTH\_MIN$  erhalten wir in der Praxis gute Linienmodelle.

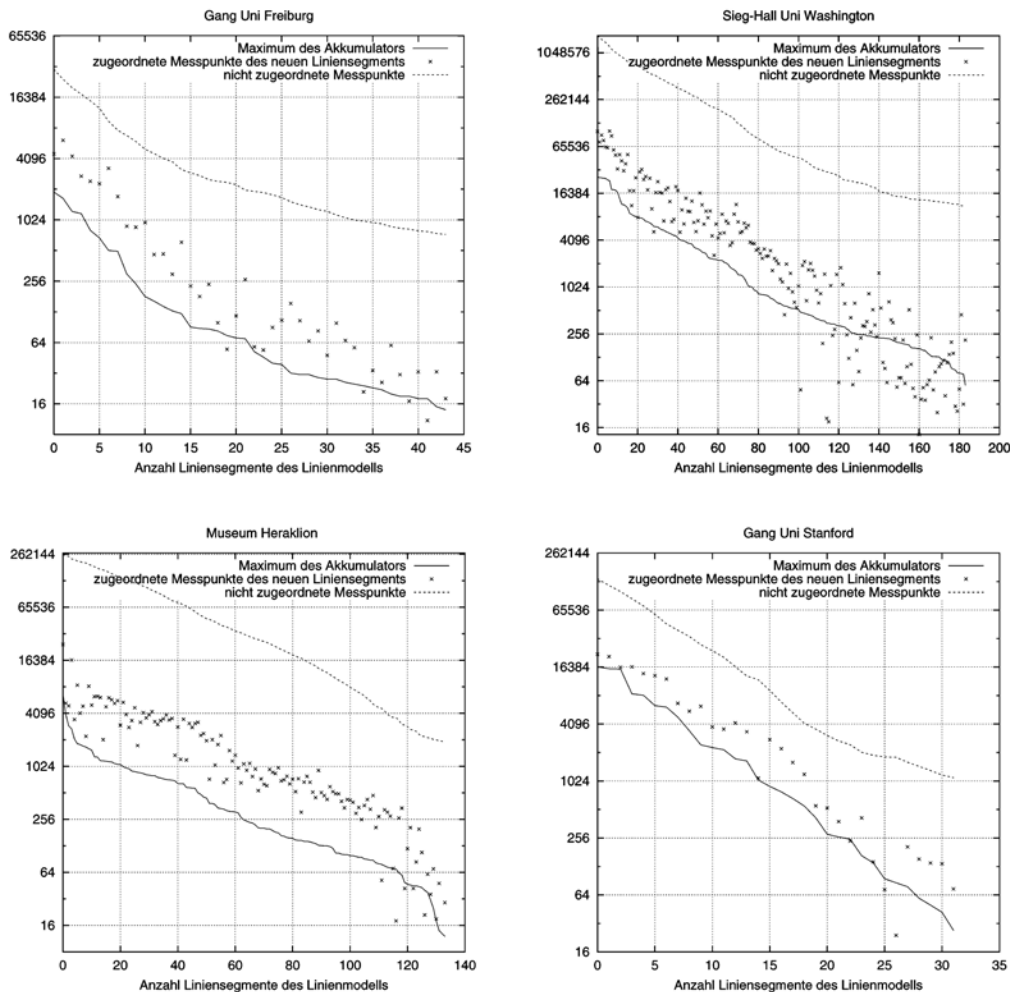
### ***AKKU\_MIN***

Dieser Parameter bildet unser Abbruchkriterium für die Extraktion von Geraden aus dem Hough-Raum. Ist das Maximum der Akkumulatormatrix kleiner als  $AKKU\_MIN$ , dann wird keine weitere Gerade extrahiert. Es ist schwierig zu beurteilen, welche Werte für diesen Parameter geeignet sind. Wir versuchen anhand der Schaubilder aus *Abbildung 5.3* einen für die Praxis geeigneten Wert für  $AKKU\_MIN$  zu finden.

In den Schaubildern ist zu erkennen, dass die Anzahl einem Liniensegment zugeordneter Messpunkte zumeist größer ist als der Akkumulatorwert der extrahierten Geraden. Dies folgt aus der Tatsache, dass einer extrahierten Geraden nicht nur die Punkte die genau auf der Geraden liegen zugeordnet werden, sondern auch jeder Messpunkt dessen Abstand zur Geraden klein genug ist. Im Verlauf des Verfahrens wird das Maximum des Akkumulators immer kleiner und die Anzahl der zugeordneten Messpunkte ist zwar nicht monoton fallend, aber wird ähnlich dem Maximum tendenziell geringer.

Der entscheidende Unterschied zwischen den Schaubildern ist jeweils der Wert des Akkumulatormaximums am Ende des Verfahrens: bei *Freiburg* 14, *Washington* 56, *Heraklion* 12, *Stanford* 27. Für uns stellt sich die Frage, ab welcher Größe des Maximums des Akkumulators die Extraktion von Geraden abgebrochen werden kann. Das größte Problem bei der Festlegung von  $AKKU\_MIN$  ist die sehr unterschiedliche Anzahl der Messpunkten der Punktemodelle von den verschiedenen Umgebungen. Die gesamte Umgebung „*Sieg-Hall Uni Washington*“ wurde mehrfach vermessen und dadurch wird ein Hindernis durch mehr Messpunkte beschrieben, als ein gleich großes Hindernis in den anderen Umgebungen, bei denen die Umgebung nicht so häufig vermessen wurde. Bei dieser Umgebung ist der kleinste Akkumulatorwert einer extrahierten Geraden viel größer als bei den anderen und wir könnten in diesem Fall  $AKKU\_MIN = 56$  wählen. Mit diesem Wert würden wir allerdings das Verfahren bei





**Abbildung 5.3:** Diese Schaubilder wurden mit dem heuristischen Verfahren von vier verschiedenen Umgebungen generiert. Sie zeigen den Verlauf des Maximums des Akkumulators, den der Anzahl der zugeordneten Punkte des neuen Liniensegments und den der Anzahl der noch nicht zugeordneten Messpunkte des Punktemodells während der Erstellung der Linienmodelle.

den anderen drei Umgebungen viel zu früh abbrechen, da jeweils der geringste Akkumulatorwert einer extrahierten Geraden kleiner 56 ist. Wir müssen uns also am kleinsten Akkumulatorwert einer extrahierten Geraden aller Umgebungen orientieren. Werte für  $AKKU\_MIN$  zwischen 10 und 15 haben sich für diese drei Umgebungen bewährt. Mit diesen Werten müssen im Beispiel „Sieg-Hall Uni Washington“ alle Geraden der Akkumulatorzellen mit Werten zwischen 56 und  $AKKU\_MIN$  extrahiert und segmentiert werden, damit die Werte dieser Zellen auf 0 gesetzt werden und somit kleiner  $AKKU\_MIN$  sind. Man könnte denken, dass dies noch einmal sehr viel Zeit in Anspruch nimmt, da bei der Zuordnung der Messpunkte zu einer Geraden alle noch

nicht zugeordneten Punkte betrachtet werden müssen. Wie man an den Kurven der nicht zugeordneten Messpunkte in den Schaubildern erkennen kann, ist diese Anzahl am Ende sehr gering im Vergleich zum Beginn des Verfahrens. Die zusätzliche Laufzeit, in der kein zulässiges Segment entsteht, ist bei kleinen Werten von  $AKKU\_MIN$  für das gesamte Verfahren nicht gravierend.

## 5.2.2 EM-Verfahren

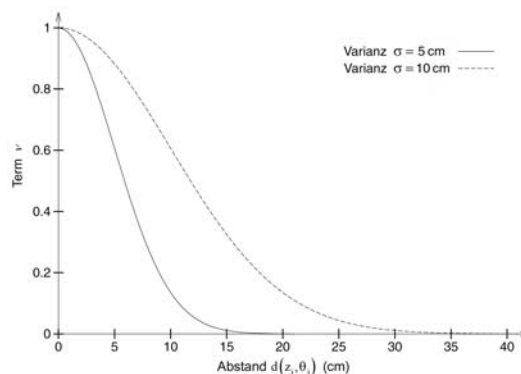
In diesem Abschnitt werden die Parameter des EM-Verfahrens aus Kapitel 4 behandelt:

### Varianz $\sigma$ :

Die Varianz  $\sigma$  spielt bei der Berechnung der Erwartungen, dass ein Punkt zu einer Geraden zugeordnet wird, eine wichtige Rolle. Wie wahrscheinlich ein Punkt einer Geraden zugeordnet wird, hängt unter anderem vom Wert dieses Parameters ab. Betrachten wir den folgenden Term (5.1) etwas genauer, welcher für die Berechnung der Erwartungen in *Abschnitt 4.3.1* benötigt wird:

$$\nu = e^{-\frac{1}{2} \frac{d^2(z_i, \theta_j^{[v]})}{\sigma^2}} \quad (5.1)$$

Wird der Abstand  $d(z_i, \theta_j^{[v]})$  eines Punktes  $z_i$  zu einer Geraden  $\theta_j^{[v]}$  größer, dann wird der ganze Term  $\nu$  dadurch kleiner und somit auch die Erwartung, dass der Punkt zur Geraden zugeordnet wird. Wie groß der Wert von  $\nu$  ist hängt aber auch von der Varianz  $\sigma$  ab, wie in *Abbildung 5.4* dargestellt.

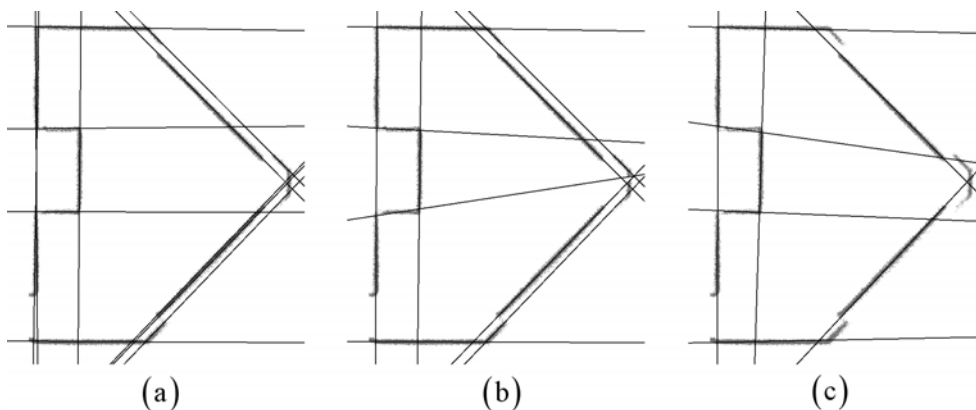


**Abbildung 5.4:** Darstellung des Terms  $\nu$  aus (5.1) mit unterschiedlichen Varianzen  $\sigma$

Bei  $\sigma = 10$  cm ist der Exponent von  $\nu$  größer als bei  $\sigma = 5$  cm, wobei der Wert des Exponenten nach oben durch 0 beschränkt ist. Somit kann der Wert von  $\nu$  nicht größer als 1 werden. Dies ist der Fall, wenn ein Punkt genau auf einer Geraden liegt und der Abstand 0 ist. Wird, wie man in *Abbildung 5.4* sieht, die Varianz erhöht, dann steigt der

maximale Abstand den ein Punkt besitzen darf, damit der Wert von  $\nu$  nicht zu gering wird. Somit werden bei einer größeren Varianz insgesamt mehr Punkte einer Geraden zugeordnet, deren Erwartung wesentlich größer als 0 ist.

Für die Festlegung des Wertes der Varianz  $\sigma$  in der Praxis betrachten wir in *Abbildung 5.5* die drei Geradenmodelle eines Raumes, welche mit unterschiedlichen Varianzen erstellt wurden. In *Abbildung 5.5 (c)* ist leicht zu erkennen, dass die Varianz  $\sigma = 15$  cm zu groß für ein gutes Geradenmodell ist, da parallel liegende Strukturen, die zu nah beisammen liegen, nur durch eine Gerade erklärt werden. Dies bewirkt, im Vergleich zu den anderen Modellen, einen höheren Wert des Gütemaßes  $G(\theta, Z)$  (vgl. *Tabelle 5.3*), da einige Messpunkte relativ weit von einer Geraden entfernt liegen. Die zwei anderen *Abbildung 5.5 (a)* und *(b)* zeigen recht ähnliche Geradenmodelle, welche die linearen Strukturen des Punktemodells sehr gut überdecken und sich im Gütemaß nur gering unterscheiden. Im Modell (a) werden jedoch einige lineare Strukturen durch mehrere Geraden beschrieben, wodurch dieses Modell mehr Geraden als notwendig besitzt. In mehreren von uns durchgeführten Experimenten hat sich herausgestellt, dass mit einer kleineren Varianz mehr Iterationen des EM-Verfahrens benötigt werden, bis ein



**Abbildung 5.5:** Ausgehend vom Punktemodell mit 41.524 Messpunkten, welches im Hintergrund zu erkennen ist, wurden Geradenmodelle gebildet, mit Varianzen  $\sigma$  von (a) 1 cm, (b) 5 cm und (c) 15 cm.

Abbildung 5.5	Varianz $\sigma$	Geraden	Iterationen	Gütemaß $G(\theta, Z)$
(a)	1 cm	12	346	70.907
(b)	5 cm	10	222	76.421
(c)	15 cm	8	145	106.665

**Tabelle 5.3:** Anzahl der Geraden, Anzahl der Iterationen und das Gütemaß der Geradenmodelle aus *Abbildung 5.5*

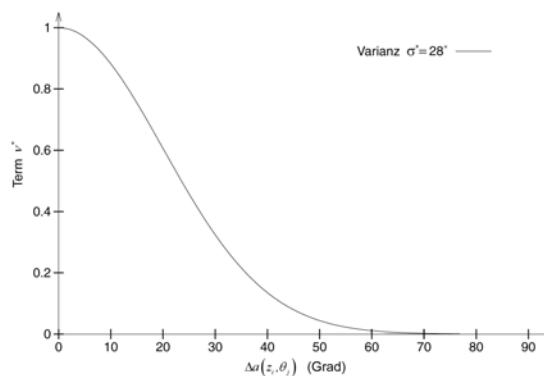
Geradenmodell konvergiert. Die *Tabelle 5.3* verdeutlicht diese Sachverhalte. In der Praxis erhalten wir mit einer Varianz  $\sigma = 5$  cm sehr gute und in der Erstellung relativ schnelle Geradenmodelle.

### Varianz $\sigma^*$ :

Die Varianz  $\sigma^*$  wird beim erweiterten Ansatz der Berechnung der Erwartungen benötigt. Geeignete Werte für diesen Parameter können relativ einfach bestimmt werden. Die Grundidee dieser Erweiterungen ist die Veränderung des Einflusses eines Punktes  $z_i$  auf eine Gerade  $\theta_j$  in Abhängigkeit von der Differenz zwischen der lokalen Ausrichtung des Punktes und der Ausrichtung der Geraden, die wir mit  $\Delta a(z_i, \theta_j)$  beschreiben. Wir wollen in der Praxis Messpunkte, die einen kleineren Wert von  $\Delta a(z_i, \theta_j)$  als ca.  $20^\circ$  besitzen, mit einer wesentlich größeren Erwartung der Geraden  $\theta_j$  zuordnen, als Messpunkte mit größeren Werten. Um dies zu realisieren muss der Wendepunkt der Kurve

$$v^* = e^{-\frac{\Delta a^2(z_i, \theta_j)}{\sigma^2}} \quad (5.2)$$

bei ca.  $20^\circ$  liegen. Belegen wir  $\sigma^*$  mit einem Wert von  $28^\circ$ , dann erhalten wir den gewünschten Verlauf der Kurve  $v^*$ , die zur Verdeutlichung in *Abbildung 5.6* dargestellt wird.



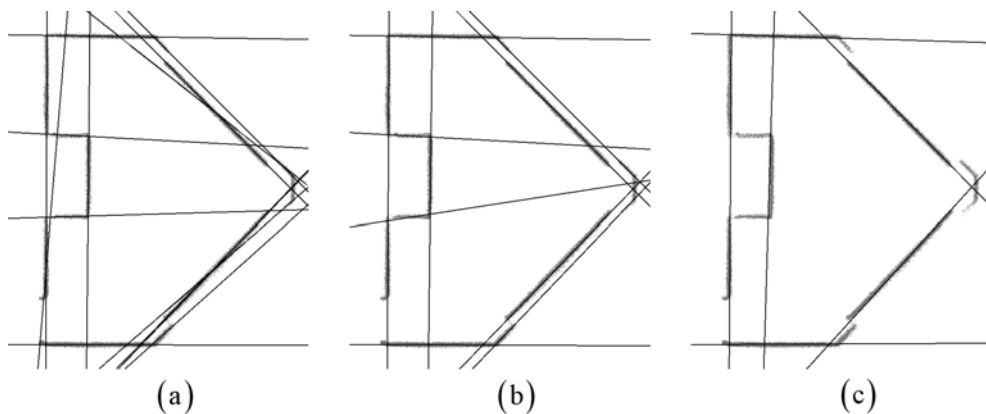
**Abbildung 5.6:** Darstellung des Terms  $v^*$  aus (5.2) mit Varianz  $\sigma^* = 28^\circ$

### **PENALTY**

Mit diesem Parameter wollen wir die Komplexität der Geradenmodelle beschränken und die Qualität der Geraden verbessern. Jede Gerade aus einem Modell soll das Gütemaß um mehr als den Wert von *PENALTY* verbessern. Die Festlegung dieses Parameters ist nicht sehr einfach und stellt das größten Probleme des gesamten EM-Verfahrens dar. Um die Schwierigkeiten der Festlegung im einzelnen zu erläutern,

betrachten wir zum einen Geradenmodelle eines Raumes und zum anderen versuchen wir mit unserem Verfahren ein simuliertes Punktemodell eines Kreises durch ein Geradenmodell zu beschreiben.

Sehen wir uns zuerst einmal die *Tabelle 5.4* und die Geradenmodelle in *Abbildung 5.7* an, die mit unterschiedlichen Werten für *PENALTY* erstellt wurden. Bei einem zu großen Wert für *PENALTY*, wie in *Abbildung 5.7 (c)*, werden Geraden die zu kurze Strukturen überdecken gelöscht, da sie das Gütemaß um weniger als *PENALTY* verbessern. Dadurch entsteht der wesentlich größere Wert des Gütemaßes im Vergleich zu den anderen Modellen. Mit unserem EM-Verfahren lassen sich kürzere lineare Strukturen schwieriger durch Geraden erklären als längere, da sie durch weniger Messpunkte beschrieben werden. Belegen wir *PENALTY* daher mit kleineren Werten, dann kann es zu Situationen wie in *Abbildung 5.7 (a)* kommen: einige lineare Strukturen werden durch mehrere Geraden überdeckt, da jede einzelne Gerade das Gütemaß des Modells um mehr als *PENALTY* verbessert und somit nicht aus dem Modell entfernt wird. Dies führt zwar zu einem geringen Wert des Gütemaßes, aber dafür steigt auch die Anzahl der Geraden. Für dieses Beispiel eines Punktemodells ist



**Abbildung 5.7:** Ausgehend vom Punktemodell mit 41.524 Messpunkten, welches im Hintergrund zu erkennen ist, wurden Geradenmodelle gebildet, mit Werten für *PENALTY* von (a) 1.000, (b) 5.000 und (c) 25.000.

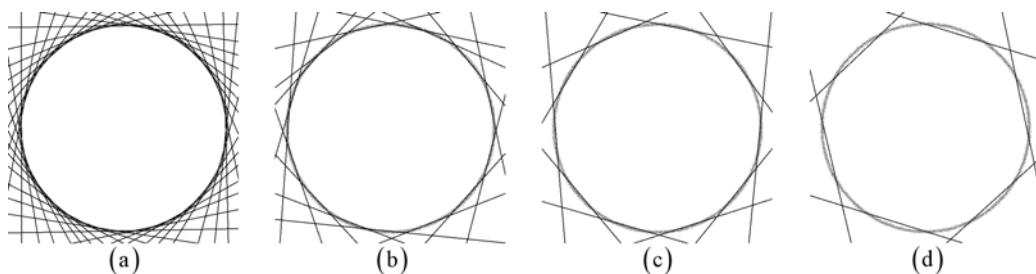
<b>Abbildung 5.7</b>	<b><i>PENALTY</i></b>	<b>Geraden</b>	<b>Gütemaß <math>G(\theta, Z)</math></b>
(a)	1.000	14	68.007
(b)	5.000	10	76.421
(c)	25.000	6	149.323

**Tabelle 5.4:** Anzahl der Geraden und das Gütemaß der Geradenmodelle aus *Abbildung 5.7*

$PENALTY = 5.000$  eine gute Wahl und wir erhalten das Geradenmodell in *Abbildung 5.7 (b)*, das fast jede lineare Struktur durch genau eine Gerade beschreibt.

Nicht bei allen Punktemodellen können wir mit diesem Wert für  $PENALTY$  ein akzeptables Ergebnis erreichen, da eine geeignete Wahl des Wertes auch von der Dichte der Messpunkte abhängt. Wird eine Umgebung mehrfach vermessen und das Punktemodell enthält mehr Messpunkte als bei einer einmaligen Vermessung, dann muss für  $PENALTY$  ein größerer Wert gewählt werden. Ein weiteres Problem bei einer generellen Festlegung dieses Parameters ist die unterschiedliche Größe der verschiedenen Punktemodelle. Bei einer sehr großen Umgebung mit mehreren Räumen kann eine Gerade, die viele Strukturen schneidet und eigentlich kein lineares Hindernis beschreibt, einen größeren Nutzen im Modell besitzen als eine Gerade, die eine kurze lineare Struktur überdeckt.

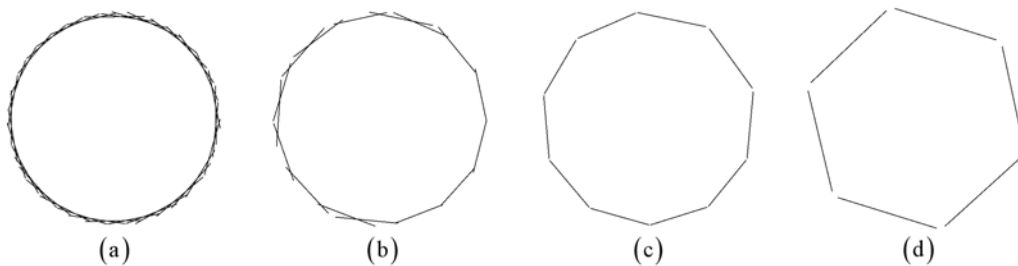
Um den Einfluss von  $PENALTY$  auf die Erstellung eines Geradenmodells weiter zu verdeutlichen, wenden wir unser Verfahren mit verschiedenen Werten für  $PENALTY$  auf ein simuliertes Punktemodell eines Kreises an. Um eine genauere Vorstellung dieser Situation zu bekommen, betrachten wir *Abbildung 5.8*.



**Abbildung 5.8:** Mit dem EM-Verfahren erstellte Geradenmodelle mit Werten für  $PENALTY$  von (a) 200, (b) 1.000, (c) 5.000 und (d) 50.000; Das zu beschreibende simulierte Punktemodell eines Kreises mit 14.400 Messpunkten wird im Hintergrund der Geradenmodelle dargestellt.

Der Wert von  $PENALTY$  beeinflusst in diesem Beispiel die Anzahl der erstellten Geraden, mit denen der Kreis beschrieben werden soll. Bei der Erstellung der Modelle wird mit einer Geraden begonnen. In Verlauf des Verfahrens nimmt die Anzahl der Geraden eines Modells zu und die optimale Anzahl der Geraden wird bestimmt, damit unser Punktemodell ausreichend gut durch das Geradenmodell erklärt wird. Bei der Suche nach der optimalen Anzahl wird der Parameter  $PENALTY$  benötigt, der den Mindestnutzen einer Geraden im Modell festlegt. Je größer der Wert für  $PENALTY$  ist, um so weniger Geraden werden für die Beschreibung des Punktemodells insgesamt eingefügt. Bei unserem Beispiel aus *Abbildung 5.8* bricht das Verfahren bei (d) am frühesten ab, und das Modell beinhaltet, im Vergleich zu den anderen Modellen, am wenigsten Geraden. Dafür verbessert am Ende jede einzelne Gerade in diesem Modell das Gütemaß sehr stark, da  $PENALTY$  einen hohen Wert besitzt. Das Punktemodell bei

(d) wird aber am ungenauesten beschrieben, da Geraden mit niedrigem Nutzen nicht zulässig sind. Das heißt, dass wir mit einem kleinen Wert für *PENALTY* ein Punktemodell genauer mit Geraden erklären können, als mit einem großen Wert, aber dafür steigt die Anzahl der Geraden im Modell (vgl. *Tabelle 5.5*). Theoretisch ist dies richtig und wir erhalten bei kleinen Werten ein sehr komplexes Geradenmodell, welches bei der anschließenden Segmentierung jedoch zu einem nicht akzeptablen Liniensmodell führen kann, da zu viele Liniensegmente entstehen. *Abbildung 5.9* und *Tabelle 5.5* verdeutlichen diesen Sachverhalt.



**Abbildung 5.9:** Liniensmodelle nach der Segmentierung der Geradenmodelle aus *Abbildung 5.8* mit Werten für *PENALTY* von (a) 200, (b) 1.000, (c) 5.000 und (d) 50.000

<b>Abbildung 5.8 bzw. Abbildung 5.9</b>	<b><i>PENALTY</i></b>	<b>Geraden bzw. Liniensegmente</b>	<b>Gütemaß <math>G(\theta, Z)</math></b>
(a)	200	36	13.842
(b)	1.000	16	27.319
(c)	5.000	10	54.625
(d)	50.000	6	136.257

**Tabelle 5.5:** Anzahl der Geraden bzw. Liniensegmente und das Gütemaß der Geradenmodelle aus *Abbildung 5.8* bzw. *Abbildung 5.9*

Mit den Beispielen dieses Abschnittes haben wir versucht, die Problematik der Festlegung des Parameters *PENALTY* zu verdeutlichen. Es ist in der Praxis sehr schwierig, einen für verschiedene Umgebungen geeigneten Wert zu finden. In zahlreichen von uns durchgeführten Experimenten haben sich Werte für *PENALTY* zwischen ca. 1.000 und 5.000 bewährt. Wir empfehlen eher einen kleinen Wert zu wählen, mit dem wir zum Teil überflüssige Liniensegmente erhalten, aber dafür ein genaueres Modell und vor allem mit nicht zu wenig Segmenten. Die meisten überflüssigen Liniensegmente können mit dem Verfahren aus *Abschnitt 4.7.1* erkannt und gelöscht werden.

## 5.3 Einfluss der lokalen Ausrichtung

Die Verwendung der lokalen Ausrichtung und den damit zusammenhängenden Einfluss auf die Ergebnisse wollen wir im Folgenden analysieren. Die zwei Verfahren zur Erstellung von Liniensegmenten werden getrennt voneinander betrachtet.

### 5.3.1 Heuristisches Verfahren

Beim heuristischen Verfahren ist die lokale Ausrichtung nicht unbedingt erforderlich um ein Linienmodell zu erstellen. Wir können aber mit der lokalen Ausrichtung die Hough-Transformation verändern und dadurch die Extraktion von Geraden aus dem Hough-Raum für unsere Situation verbessern. Wie bereits in *Abschnitt 3.1.3* beschrieben, ist ein Vorteil bei der Berücksichtigung der lokalen Ausrichtung die geringere Anzahl der zu in- bzw. dekrementierenden Zellen der Akkumulatormatrix, die beim Einfügen bzw. Entfernen eines Messpunktes aus dem Hough-Raum berücksichtigt werden müssen.

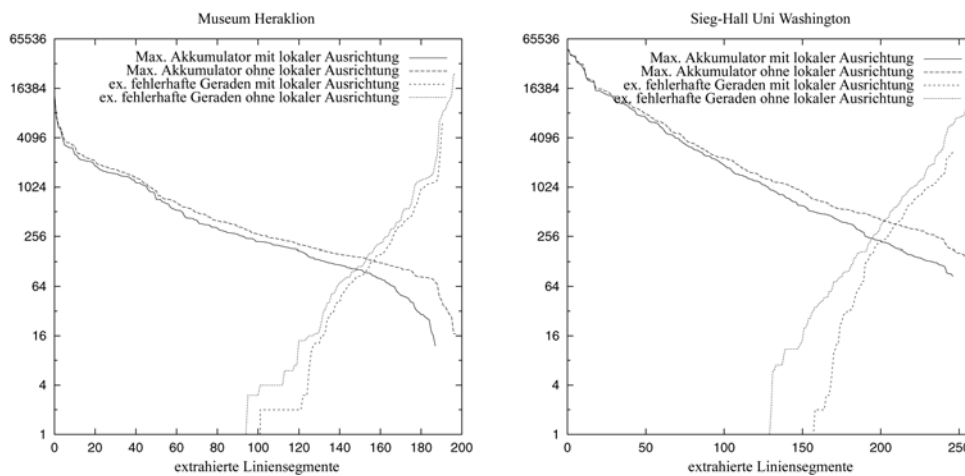
Wir vergleichen nun die Ergebnisse des heuristischen Verfahrens mit und ohne Erweiterung durch die lokale Ausrichtung aus *Abschnitt 3.1.3*. Hierfür verwenden wir zwei verschiedene Punktemodelle, die in Linienmodelle umgewandelt wurden. Bei der Erstellung der Linienmodelle ergaben sich die in *Abbildung 5.10* gezeigten Kurven, die zum einen den Verlauf des Maximums des Akkumulators der extrahierten Geraden darstellen und die zum anderen die Anzahl der aus dem Hough-Raum extrahierten *fehlerhaften Geraden*<sup>7</sup> aufzeigen. Man kann erkennen, dass mit Verwendung der lokalen Ausrichtung weniger fehlerhafte Geraden extrahiert werden. Diese Anzahl dieser Geraden nimmt bei beiden Ansätzen ungefähr gleich schnell zu, aber mit der lokalen Ausrichtung wird die erste fehlerhafte Gerade zu einem späteren Zeitpunkt des Verfahrens extrahiert. Mit weniger fehlerhaften Geraden müssen auch weniger Segmentierungen solcher Geraden durchgeführt werden. Des Weiteren ist zu erkennen, dass der Gebrauch der lokalen Ausrichtung zu geringeren Maxima im Akkumulator bei der Extraktion der Geraden führt. Diese beruht auf dem geringeren Wertebereich des Winkels bei der Hough-Transformation mit lokaler Ausrichtung. Wie bereits in *Abschnitt 3.1.3* erläutert, kann mit einem geringeren Wertebereich die Laufzeit der Berechnung des Akkumulators verkürzt werden.

Unter Berücksichtigung dieser Aspekte betrachten wir nun die erstellten Linienmodelle mit und ohne Verwendung der lokalen Ausrichtung in *Abbildung 5.11* bzw. *5.12* und *Tabelle 5.6* bzw. *5.7*. Die Punktemodelle werden durch das heuristische Verfahren mit

---

<sup>7</sup> Eine *fehlerhafte Gerade* ist eine aus dem Hough-Raum extrahierte Gerade, aus der mit der Segmentierung kein zulässiges Liniensegment entsteht.





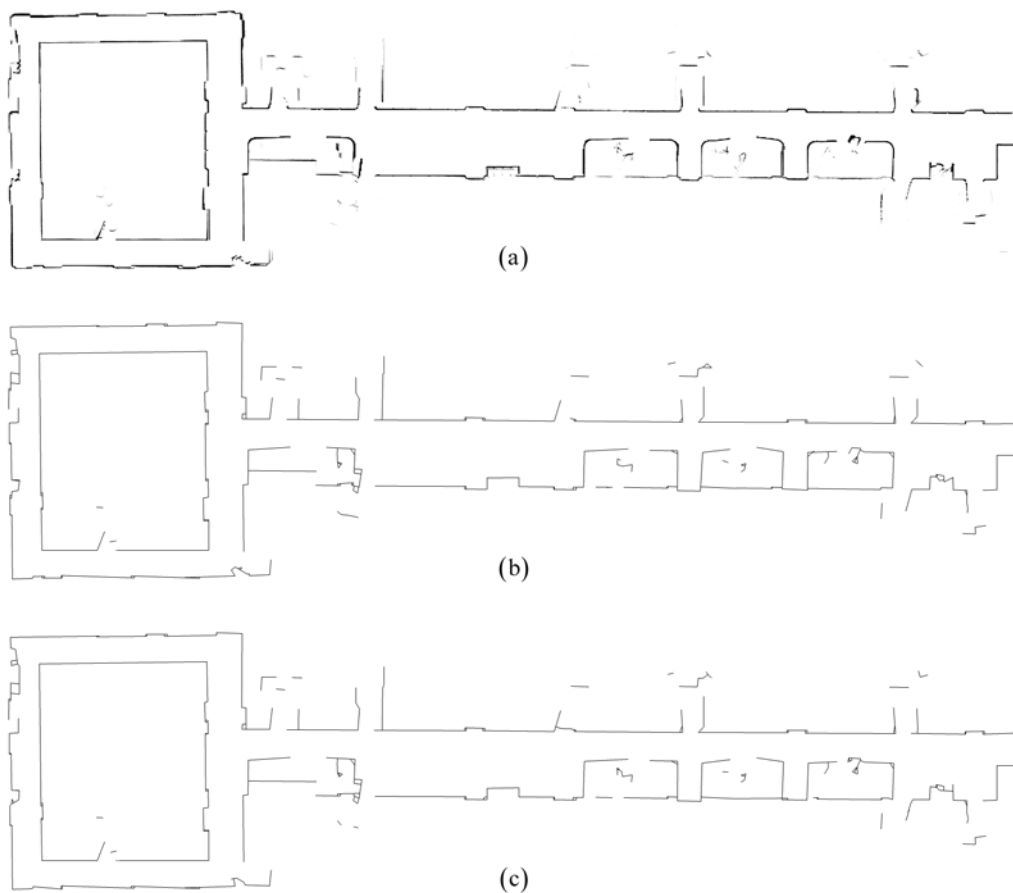
**Abbildung 5.10:** Einfluss der Verwendung der lokalen Ausrichtung auf den Verlauf des heuristischen Verfahrens; Die Ordinate gibt zum einen das Maximum des Akkumulators der extrahierten Geraden und zum anderen die Anzahl der extrahierten fehlerhaften Geraden an. Auf der Abszisse ist die Anzahl der extrahierten Liniensegmente des Linienmodells aufgetragen.

und ohne Verwendung der lokalen Ausrichtung in sehr gute Linienmodelle umgewandelt. Die Umgebung aus *Abbildung 5.11* bzw. *5.12* besitzt eine Größe von  $53\text{ m} \times 14\text{ m}$  bzw.  $38\text{ m} \times 18\text{ m}$  und wurde anhand von 1.829.645 bzw. 311.823 Messpunkten vermessen. Mit unserem Verfahren wurde das jeweilige Punktemodell mit (ohne) Benutzung der lokalen Ausrichtung in ein Linienmodell mit 296 (297) bzw. 199 (196) Liniensegmente umgewandelt. Wir erhalten exakte und sehr kompakte Linienmodelle der Umgebungen.

Vergleichen wir die Ergebnisse mit und ohne Verwendung der lokalen Ausrichtung, dann werden ca. 5% mehr Liniensegmente aus dem Hough-Raum extrahiert, wenn die lokale Ausrichtung nicht berücksichtigt wird. Nach der Vereinigung der Liniensegmente ist die Anzahl der Segmente bei beiden Ansätzen fast identisch. Dies lässt darauf schließen, dass die zusätzlich extrahierten Segmente ohne Zuhilfenahme der lokalen Ausrichtung zumeist mit der Vereinigung der Liniensegmente wieder entfernt werden. Wir können also mit Verwendung der lokalen Ausrichtung die Anzahl der extrahierten Liniensegmente verringern und erhalten am Ende des Verfahrens trotzdem ein Modell, welches keineswegs schlechter als das Modell ohne lokaler Ausrichtung ist.

Bei einem genaueren visuellen Vergleich der beiden Ansätze sind kaum Unterschiede in den Modellen zu erkennen. Alle großen und signifikanten linearen Strukturen der jeweiligen Umgebung werden in beiden Verfahren sehr genau durch Liniensegmente beschrieben. Nur bei zumeist etwas kürzeren Segmenten können kleine Unterschiede in den Modellen erkannt werden, die für das Gesamtergebnis allerdings nicht sehr gravierend sind. Die Tatsache, dass wir die Laufzeit des heuristischen Verfahrens mit Verwendung der lokalen Ausrichtung verringern, spricht für die Erweiterung der

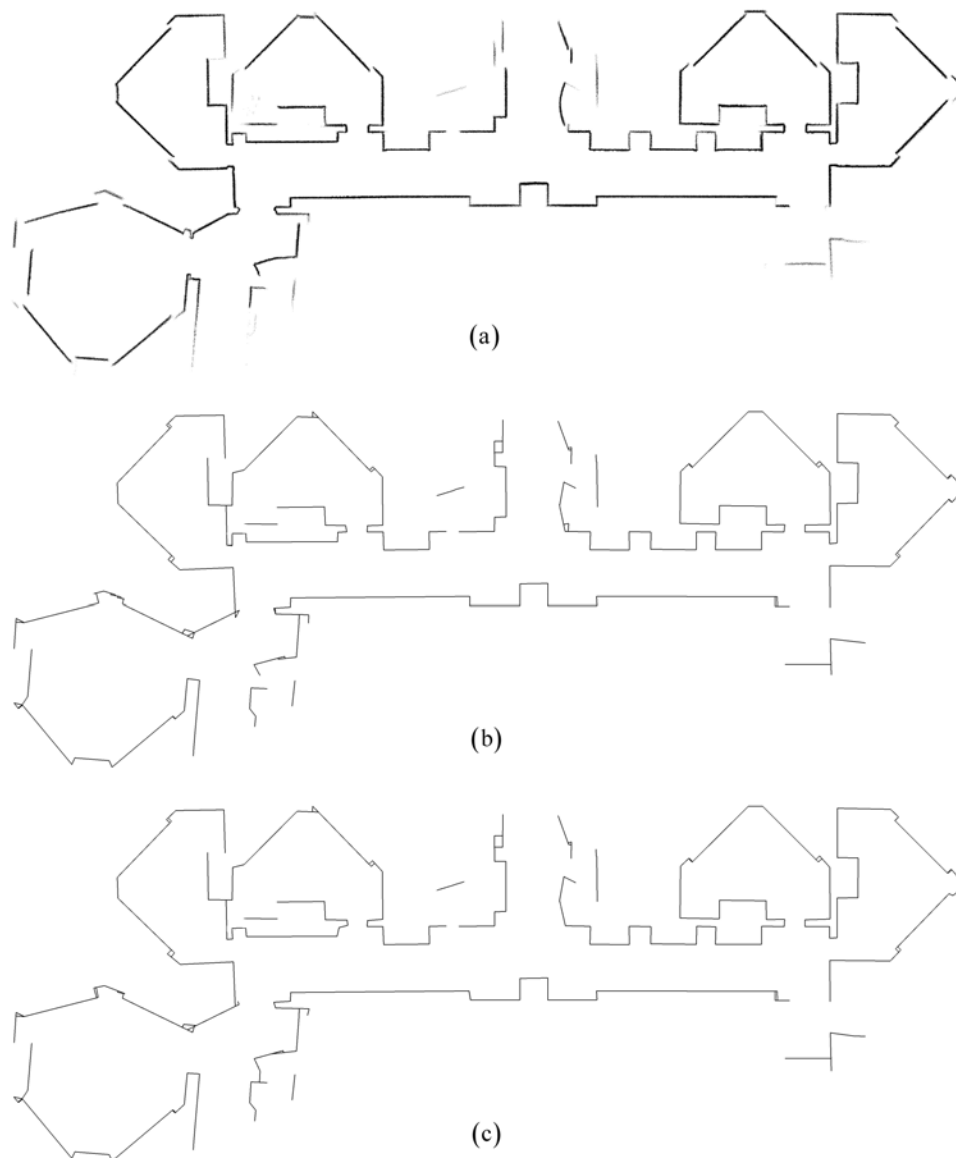
Hough-Transformation durch die lokale Ausrichtung. In den Beispielen aus *Abbildung 5.11* und *5.12* haben wir den Wertebereich des Winkels bei der Transformation der Messpunkte in den Hough-Raum auf  $40^\circ$  beschränkt und somit die Berechnung der Akkumulatormatrix um das  $180^\circ/40^\circ = 4,5$ -fache beschleunigt.



**Abbildung 5.11:** Modelle von der Sieg-Hall Uni Washington ( $53\text{ m} \times 14\text{ m}$ ); Ausgehend vom Punktemodell (a) mit 1.829.645 Messpunkten wurden Linienmodelle gebildet, (b) mit bzw. (c) ohne Berücksichtigung der lokalen Ausrichtung bei der Berechnung des Akkumulators.

Abbildung 5.11	Lokale Ausrichtung	Segmente aus Hough-Raum	Segmente nach Vereinigung	mit „kurzen“ Segmenten
(b)	mit	252	175	296
(c)	ohne	265	177	297

**Tabelle 5.6:** Anzahl der Liniensegmente der zwei Linienmodelle aus *Abbildung 5.11* an verschiedenen Stellen des Verfahrens



**Abbildung 5.12:** Modelle von einem Museum in Heraklion ( $38\text{ m} \times 18\text{ m}$ ); Ausgehend vom Punktemodell (a) mit 311.823 Messpunkten wurden die Linienmodelle gebildet, (b) mit bzw. (c) ohne Berücksichtigung der lokalen Ausrichtung bei der Berechnung des Akkumulators.

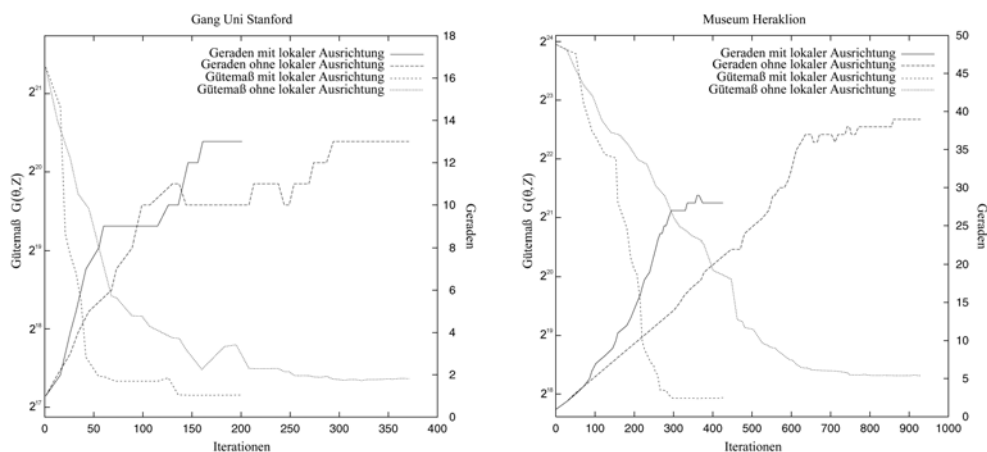
Abbildung 5.12	Lokale Ausrichtung	Segmente aus Hough-Raum	Segmente nach Vereinigung	mit „kurzen“ Segmenten
(b)	mit	187	140	199
(c)	ohne	196	141	196

**Tabelle 5.7:** Anzahl der Liniensegmente der zwei Linienmodelle aus *Abbildung 5.12* an verschiedenen Stellen des Verfahrens

### 5.3.2 EM-Verfahren

Die lokale Ausrichtung können wir im EM-Verfahren an zwei verschiedenen Stellen einsetzen. Beim Einfügen einer neuen Geraden in das momentane Geradenmodell kann die lokale Ausrichtung verwendet werden, um nicht zufällig die Ausrichtung der neuen Geraden festzulegen (*Abschnitt 4.4*). Des Weiteren gibt es die Möglichkeit, die lokale Ausrichtung bei der Berechnung der Erwartungen im E-Schritt zu berücksichtigen (*Abschnitt 4.3.1*), damit die Erwartungen nicht allein von den Abständen der Messpunkte zu den Geraden abhängig sind.

Vergleichen wir zunächst die Ergebnisse der erstellten Geraden- und Linienmodelle mit und ohne Verwendung der lokalen Ausrichtung beim Einfügen einer neuen Geraden. Für den Vergleich der beiden Ansätze haben wir zwei verschiedene Punktemodelle in Geraden- und Linienmodelle umgewandelt. *Abbildung 5.13* zeigt den Verlauf des Gütemaßes und die Anzahl der verwendeten Geraden während der Suche nach der optimalen Anzahl von Geraden bei der Erstellung der Geradenmodelle. In beiden Schaubildern ist zu erkennen, dass deutlich weniger Iterationen bei der Suche nach der optimalen Anzahl der Geraden für das Geradenmodell benötigt werden, wenn die lokale Ausrichtung beim Einfügen einer neuen Geraden verwendet wird. Dies ist durch die schnellere Anpassung der neu eingefügten Geraden an das Punktemodell zu erklären, da die lokale Ausrichtung eines Punktes Informationen über die mögliche Lage eines linearen Hindernisses beinhaltet. Mit anderen Worten heißt das, dass durch die lokale Ausrichtung die neuen Geraden bei der Optimierung des Geradenmodells weniger verändert werden müssen. Hiermit lässt sich auch der unterschiedliche Verlauf des Gütemaßes erklären. Des Weiteren ist in beiden Schaubildern das Gütemaß durch die



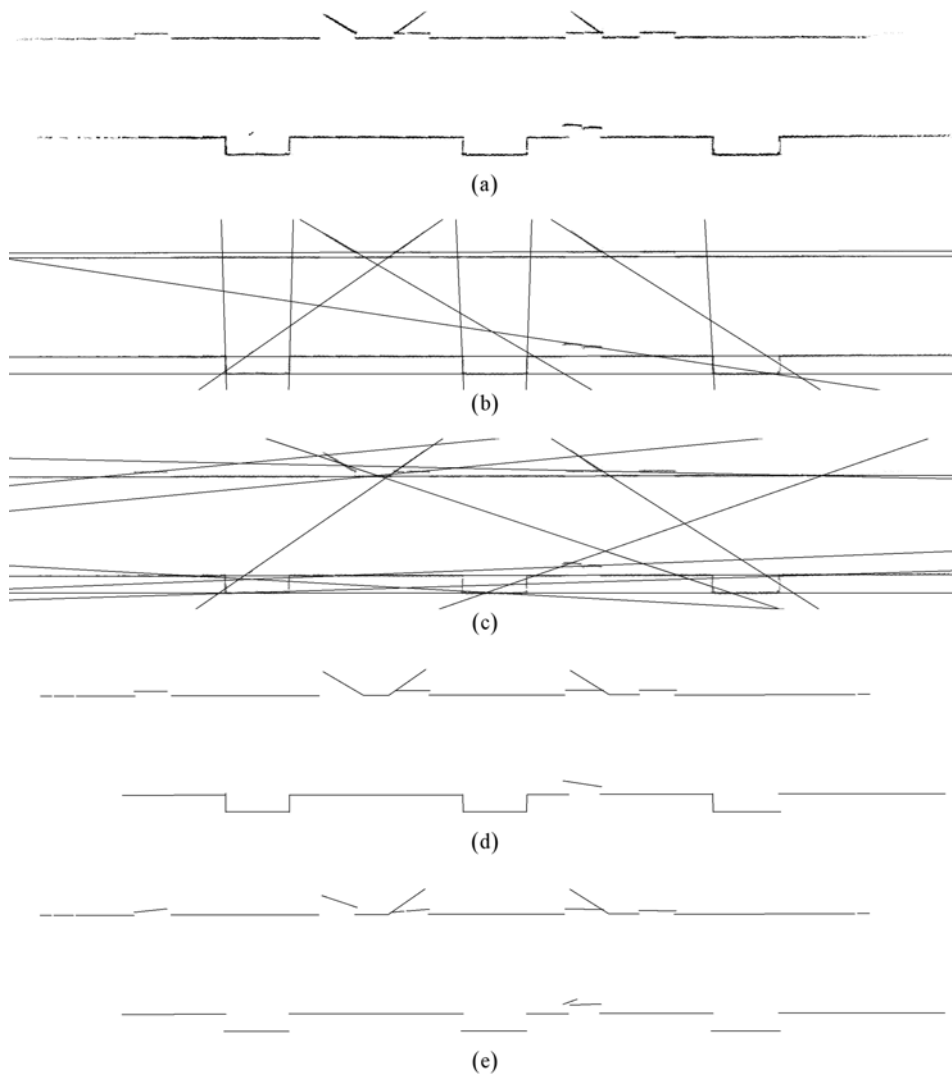
**Abbildung 5.13:** Einfluss der Verwendung der lokalen Ausrichtung bei den neu eingefügten Geraden auf den Verlauf des EM-Verfahrens; Die Schaubilder stellen die Entwicklung des Gütemaßes und die der Anzahl der Geraden während der Erstellung der Geradenmodelle von zwei verschiedenen Umgebungen dar.

Benutzung der lokalen Ausrichtung am Ende des Verfahrens kleiner. Wie bereits im *Abschnitt 5.1* erläutert, deutet ein geringerer Wert für das Gütemaß auf ein besseres Geradenmodell hin, wobei die Anzahl der verwendeten Geraden mit berücksichtigt werden muss. Im Beispiel des Museums in Heraklion ist deutlich zu erkennen, dass ohne Verwendung der lokalen Ausrichtung die Anzahl der Geraden wesentlich größer ist. Insgesamt können wir somit auf bessere Geradenmodelle mit Verwendung der lokalen Ausrichtung schließen.

Für die Bestätigung dieser Vermutung betrachten wir nun die *Abbildung 5.14* und *5.15* mit der jeweils dazugehörenden *Tabelle 5.8* und *5.9*. In diesen Beispielen wird deutlich, dass wir in unserem EM-Verfahren auf die Verwendung der lokalen Ausrichtung nicht verzichten können. Die Unterschiede zwischen den Geradenmodellen in den *Abbildungen (b)* und *(c)* sind zum Teil sehr gravierend. Es ist gut zu erkennen, dass ohne Gebrauch der lokalen Ausrichtung nur die sehr langen linearen Strukturen in den Punktemodellen ausreichend gut durch jeweils eine Gerade überdeckt werden. Viele etwas kürzere Strukturen werden entweder gar nicht oder nur sehr schlecht durch zumeist mehrere Geraden überdeckt. Wir erhalten deshalb nach der anschließenden Segmentierung nicht besonders gute Linienmodelle, wie in den *Abbildungen (e)* zu erkennen ist. Mit Verwendung der lokalen Ausrichtung beim Erstellen der Geradenmodelle können wir die Ergebnisse wesentlich verbessern.

Die Umgebung der *Abbildung 5.14* besitzt eine Größe von  $17\text{ m} \times 5\text{ m}$  und wurde mittels 162.554 Messpunkten vermessen. Das Punktemodell *(a)* konnte mit unserem EM-Verfahren unter Verwendung der lokalen Ausrichtung durch das Geradenmodell *(b)* mit 13 Geraden erklären werden und mit Hilfe der anschließenden Segmentierung in das Linienmodell *(d)* mit 30 Segmenten umgewandelt werden. Das Ergebnis ist bei dieser einfacheren Umgebung sehr gut.

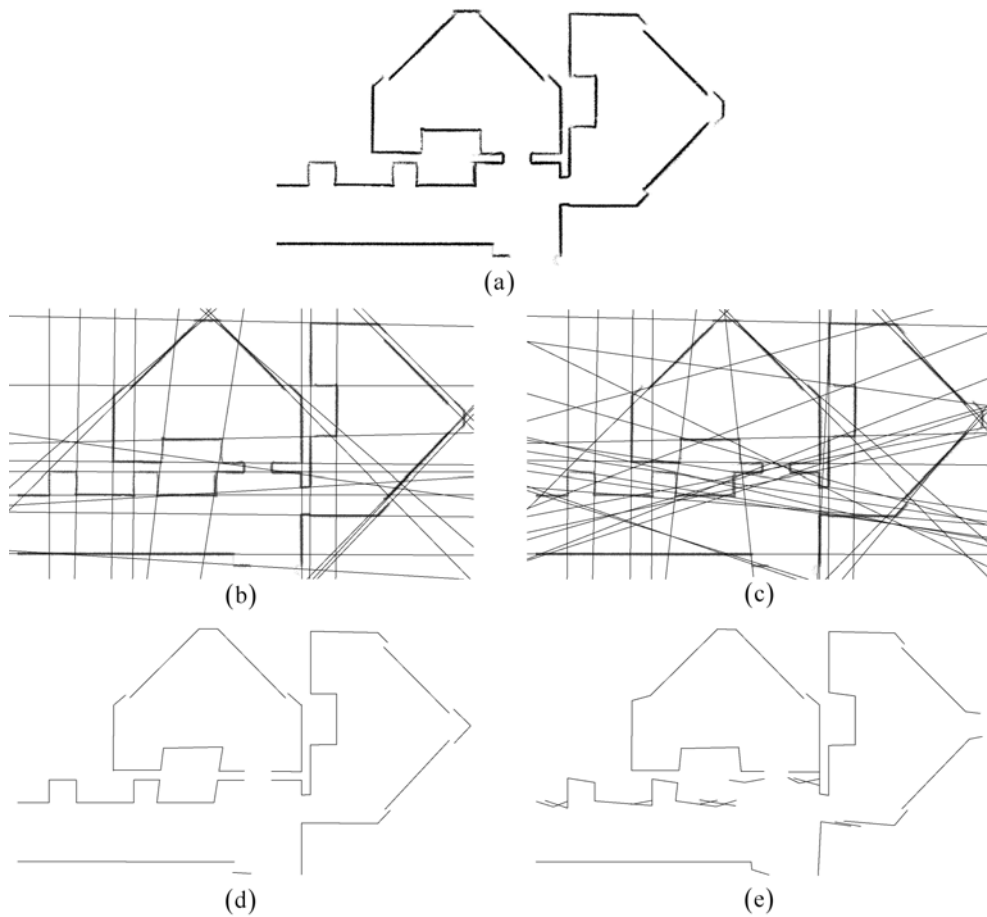
Betrachten wir hingegen die Ergebnisse einer komplexeren Umgebung in *Abbildung 5.15 (b)* und *(d)*, dann wird das größte Problem dieses Verfahrens deutlich. Da wir für die Beschreibung unseres Punktemodells im EM-Ansatz zuerst Geraden und nicht Segmente benutzen, können mehrere unterschiedliche lineare Strukturen durch nur eine einzige Gerade beschrieben werden. Sind diese Strukturen nicht alle exakt gleich ausgerichtet oder liegen nicht genau auf einer Linie, dann können nicht optimal an den Hindernissen ausgerichtete Liniensegmente entstehen, wie in *(d)* zu erkennen ist. In diesem Beispiel wurde eine Umgebung der Größe  $14\text{ m} \times 9\text{ m}$  mit 133.134 Messpunkten vermessen. Wir konnten mit unserem EM-Verfahren aus dem Punktemodell *(a)* das Geradenmodell *(b)* mit 28 Geraden und das Linienmodell *(d)* mit 43 Segmenten erstellen. Bei der Betrachtung der Zahlen in *Tabelle 5.9* wird klar, dass viele einzelne Geraden zu mehreren Liniensegmenten umgewandelt wurden. Dadurch können die so eben beschriebenen Fehler in der Lage der Liniensegmente entstehen. Trotz dieses kleinen Mankos ist das Ergebnis dieses Linienmodells mit Verwendung der lokalen Ausrichtung gut einzuschätzen.



**Abbildung 5.14:** Modelle von einem Gang eines Gebäudes der Uni Stanford ( $17\text{ m} \times 5\text{ m}$ ); Ausgehend vom Punktemodell (a) mit 162.554 Messpunkten wurden Geradenmodelle gebildet, (b) mit bzw. (c) ohne Berücksichtigung der lokalen Ausrichtung beim Einfügen neuer Geraden. Das Linienmodell (d) bzw. (e) wurde aus dem Geradenmodell (b) bzw. (c) gebildet.

Abbildung 5.14	Lokale Ausrichtung	Geraden	Segmente aus Geradenmodell	Segmente nach Löschen
(b) und (d)	mit	13	31	30
(c) und (e)	ohne	13	38	27

**Tabelle 5.8:** Anzahl der Geraden der Geradenmodelle und Anzahl der Liniensegmente an verschiedenen Stellen des Verfahrens der Linienmodelle aus *Abbildung 5.14*

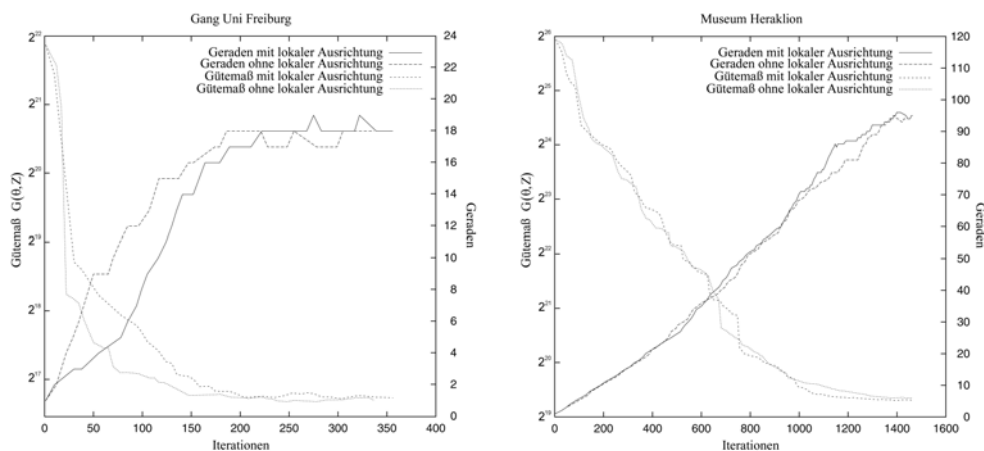


**Abbildung 5.15:** Modelle von einem Teil eines Museums in Heraklion ( $14\text{ m} \times 9\text{ m}$ ); Ausgehend vom Punktemodell (a) mit 133.134 Messpunkten wurden Geradenmodelle gebildet, (b) mit bzw. (c) ohne Berücksichtigung der lokalen Ausrichtung beim Einfügen neuer Geraden. Das Linienmodell (d) bzw. (e) wurde aus dem Geradenmodell (b) bzw. (c) gebildet.

Abbildung 5.15	Lokale Ausrichtung	Geraden	Segmente aus Geradenmodell	Segmente nach Löschen
(b) und (d)	mit	28	50	43
(c) und (e)	ohne	39	73	55

**Tabelle 5.9:** Anzahl der Geraden der Geradenmodelle und Anzahl der Liniensegmente an verschiedenen Stellen des Verfahrens der Linienmodelle aus *Abbildung 5.15*

Als Nächstes wollen wir die Verwendung der lokalen Ausrichtung bei der Berechnung der Erwartungen im E-Schritt analysieren. Wie bereits in *Abschnitt 4.3.1* erläutert wollen wir, beim erweiterten Ansatz der Berechnung der Erwartungen, nicht nur den Abstand der Messpunkte zu den Geraden, sondern auch den Unterschied zwischen der lokalen Ausrichtung der Messpunkte und der Ausrichtung der Geraden berücksichtigen. Für den Vergleich der beiden Ansätze haben wir zwei verschiedene Punktemodelle in Geraden- und Linienmodelle umgewandelt. *Abbildung 5.16* zeigt den Verlauf des Gütemaßes und der Anzahl der verwendeten Geraden während der Suche nach der optimalen Anzahl der Geraden bei der Erstellung der Geradenmodelle. In den Schaubildern unterscheidet sich der Verlauf der Kurven mit und ohne Verwendung der lokalen Ausrichtung nur sehr gering. Bei der Erstellung der Geradenmodelle in den Beispielen werden mit und ohne lokaler Ausrichtung ungefähr gleich viel Iterationen benötigt. Die Unterschiede im Gütemaß und der Anzahl der Geraden sind am Ende des Verfahrens nur sehr gering. Das heißt, dass wir anhand dieser Daten keine entscheidende Beeinflussung unseres EM-Verfahrens durch den Gebrauch der lokalen Ausrichtung bei der Berechnung der Erwartungen feststellen können. Aus dieser Tatsache können wir auf sehr ähnliche Geradenmodelle der beiden Ansätze schließen.

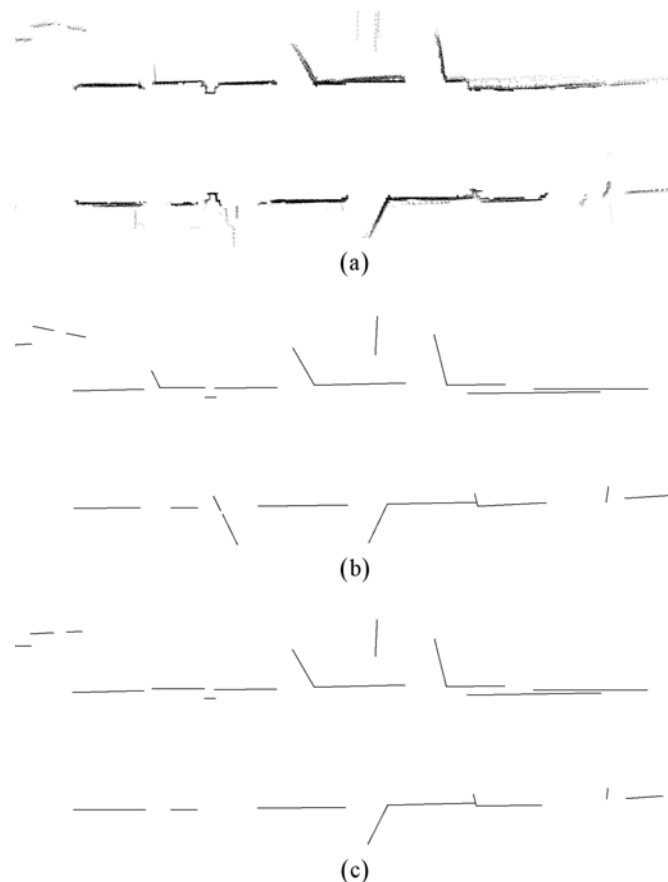


**Abbildung 5.16:** Einfluss der Verwendung der lokalen Ausrichtung bei der Berechnung der Erwartungen auf den Verlauf des EM-Verfahrens; Die Schaubilder stellen die Entwicklung des Gütemaßes und der Anzahl der Geraden während der Erstellung der Geradenmodelle von zwei verschiedenen Umgebungen dar.

Betrachten wir nun die *Abbildung 5.17* und *5.18* mit der jeweils dazugehörigen *Tabelle 5.10* und *5.11*. In diesen Beispielen wird deutlich, dass die Verwendung der lokalen Ausrichtung im E-Schritt unsere Ergebnisse des EM-Verfahrens kaum verändert. Es sind zwar einige Unterschiede in den Abbildungen (b) und (c) der Linienmodelle zu erkennen, welche zum größten Teil aber nicht durch die Verwendung der lokalen Ausrichtung entstehen. Da wir beim Einfügen einer neuen Geraden einen



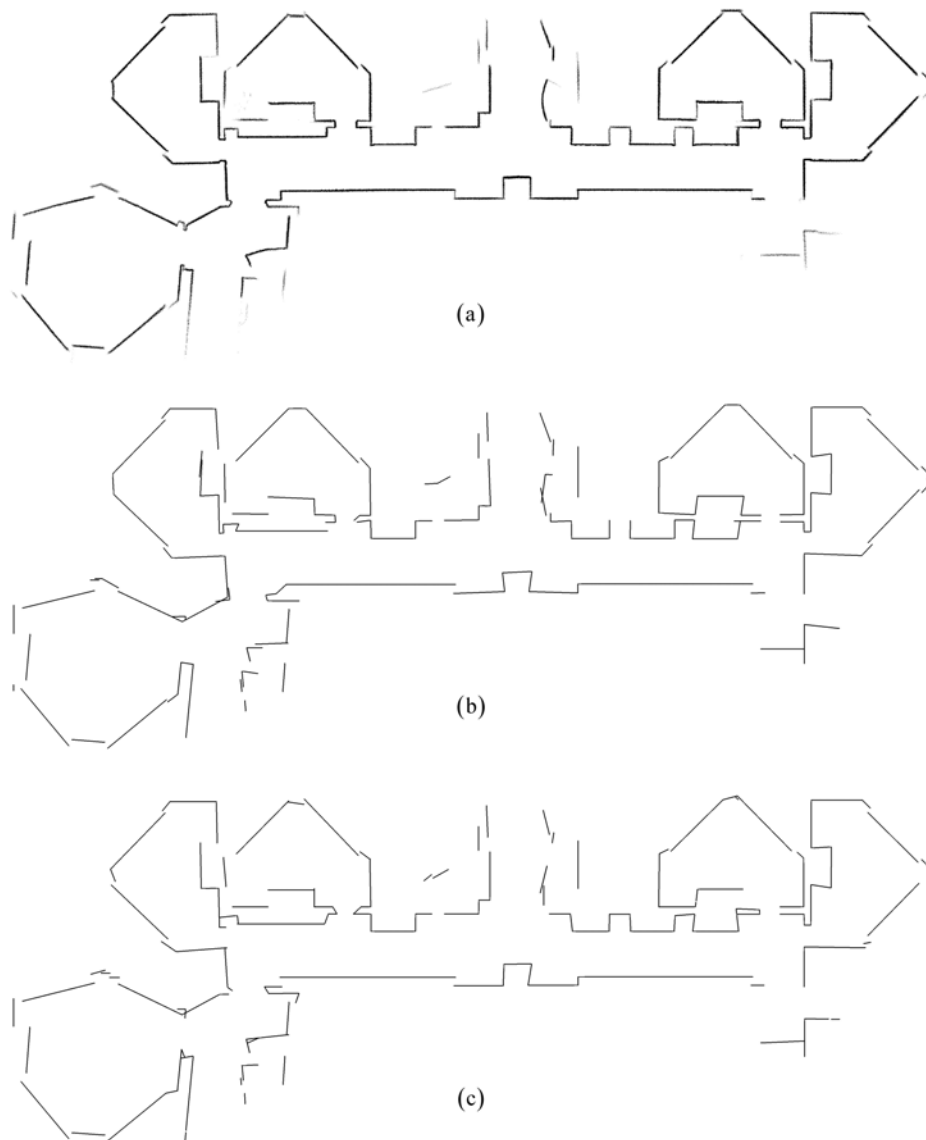
nicht gut erklärten Messpunkt zufällig auswählen, auf dem die neue Gerade liegen soll, beinhaltet unser EM-Verfahren eine Zufallskomponente. Deshalb ist es sehr schwierig, den Einfluss der lokalen Ausrichtung auf unser Verfahren zu analysieren, da die Ergebnisse immer einen gewissen Zufall beinhalten. Trotzdem ist festzustellen, dass die lokale Ausrichtung unsere Ergebnisse auf keinen Fall verschlechtert.



**Abbildung 5.17:** Modelle von einem Gang eines Gebäudes der Uni Freiburg (13 m × 5 m); Ausgehend vom Punktemodell (a) mit 35.399 Messpunkten wurden Linienmodelle gebildet, (b) mit bzw. (c) ohne Berücksichtigung der lokalen Ausrichtung bei der Berechnung der Erwartungen im E-Schritt.

Abbildung 5.17	Lokale Ausrichtung	Geraden	Segmente aus Geradenmodell	Segmente nach Löschen
(b)	mit	18	46	29
(c)	ohne	18	47	27

**Tabelle 5.10:** Anzahl der Geraden der Geradenmodelle und Anzahl der Liniensegmente an verschiedenen Stellen des Verfahrens der Linienmodelle aus *Abbildung 5.17*



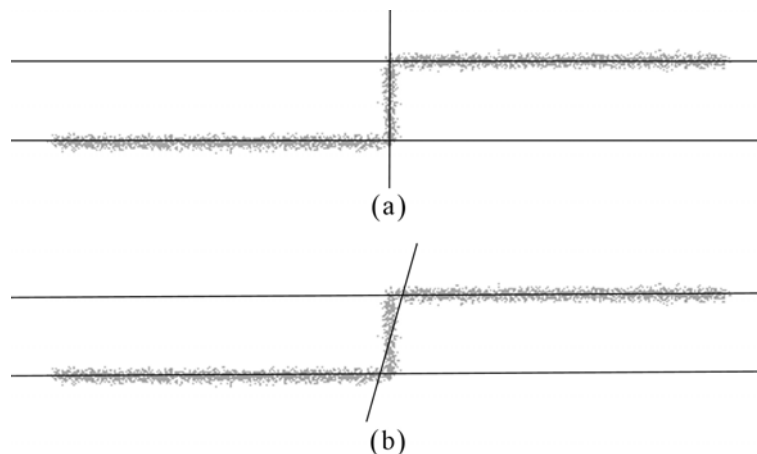
**Abbildung 5.18:** Modelle von einem Museum in Heraklion ( $38\text{ m} \times 18\text{ m}$ ); Ausgehend vom Punktemodell (a) mit 311.823 Messpunkten wurden Linienmodelle gebildet, (b) mit bzw. (c) ohne Berücksichtigung der lokalen Ausrichtung bei der Berechnung der Erwartungen im E-Schritt.

Abbildung 5.18	Lokale Ausrichtung	Geraden	Segmente aus Geradenmodell	Segmente nach Löschen
(b)	mit	95	217	139
(c)	ohne	94	240	140

**Tabelle 5.11:** Anzahl der Geraden der Geradenmodelle und Anzahl der Liniensegmente an verschiedenen Stellen des Verfahrens der Linienmodelle aus *Abbildung 5.18*

In unseren zahlreichen Versuchen haben sich an manchen Stellen kleine Vorteile durch die lokale Ausrichtung bei der Optimierung der Geraden ergeben. Überdeckt zum Beispiel mit Verwendung der lokalen Ausrichtung eine Gerade sehr viele Hindernisse, dann werden Messpunkte von Hindernissen mit ähnlicher Ausrichtung zur Geraden mit einer größeren Erwartung dieser Geraden zugeordnet, als Messpunkte von Hindernissen, die fast senkrecht zur Geraden liegen. Diese Situation tritt vor allem bei größeren und komplexeren Umgebungen auf und wir können durch die Verwendung der lokalen Ausrichtung die Geraden des Modells besser optimieren.

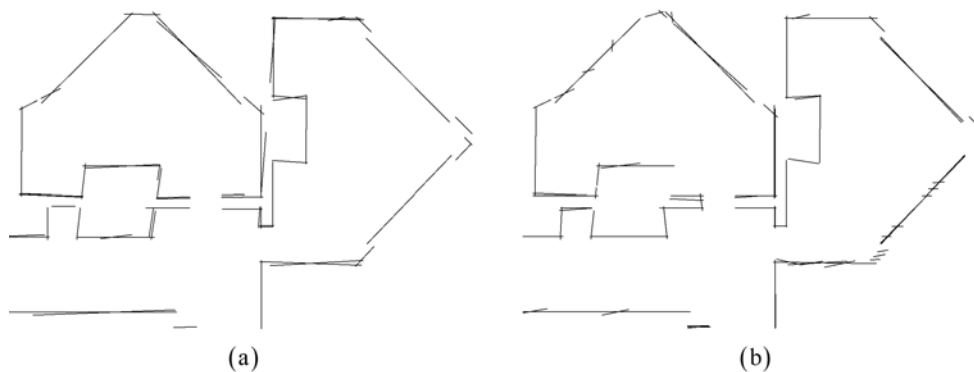
Auch bei der Anpassung einer Geraden an eine nicht sehr lange Struktur, die durch eine relativ geringe Anzahl von Messpunkten beschrieben wird, ist die Verwendung der lokalen Ausrichtung vorteilhaft, wie in *Abbildung 5.19* dargestellt. In diesem Beispiel ist zu erkennen, dass ohne die lokale Ausrichtung die Gerade, welche die senkrechte Struktur beschreiben soll, durch die zwei waagerechten Strukturen beeinflusst und somit nicht optimal angepasst wird. Die Geraden, welche die waagerechten Strukturen beschreiben sollen, werden zwar durch die senkrechte Struktur auch beeinflusst, aber dieser Einfluss ist verhältnismäßig gering, da die Anzahl der Messpunkte je waagerechter Struktur groß genug ist. Mit der lokalen Ausrichtung kann ein „fehlerhafter“ Einfluss von Messpunkten auf eine Gerade verringert bzw. vermieden werden, wie in *Abbildung 5.19 (a)*. Dies ist jedoch nur dann möglich, wenn die Differenz zwischen den Ausrichtungen der verschiedenen Strukturen groß genug ist.



**Abbildung 5.19:** Im Hintergrund ist ein simuliertes Punktemodell mit 4.500 Punkten und einer Größe von  $400 \text{ cm} \times 60 \text{ cm}$  zu sehen. Ausgehend vom Punktemodell wurden Linienmodelle gebildet, (a) mit bzw. (b) ohne Berücksichtigung der lokalen Ausrichtung bei der Berechnung der Erwartungen im E-Schritt.

Ein weiterer Vorteil ergibt sich beim Übergang vom Geradenmodell zum Linienmodell: bei der Segmentierung der Geraden werden die Messpunkte besser den Geraden zugeordnet und es entstehen weniger überflüssige Liniensegmente. Dies ist zum einen

an der Anzahl der erstellten Segmente in *Tabelle 5.11* zu erkennen; die Geradenmodelle besitzen fast gleich viele Geraden, doch mit Verwendung der lokalen Ausrichtung werden ca. 10% weniger Segmente gebildet. Nach dem Löschen von überflüssigen Liniensegmenten ist die Anzahl der Segmente in den beiden Modellen wieder fast identisch. Zur Verdeutlichung zeigen wir in *Abbildung 5.20* einen kleinen Ausschnitt der Modelle vom Museum in Heraklion aus *Abbildung 5.18* direkt nach der Segmentierung der Geradenmodelle. Es ist klar zu erkennen, dass ohne Verwendung der lokalen Ausrichtung einige zumeist kurze Liniensegmente erstellt werden, deren Ausrichtung fehlerhaft ist. Ein solches, zu einer Struktur fehlerhaft ausgerichtetes Segment, entsteht aus einer Geraden, welche ein ganz anderes Hindernis in der Umgebung erklärt. Jedoch nicht alle fehlerhaften Segmente können mit der lokalen Ausrichtung vermieden werden. In *Abbildung 5.20 (a)* werden die meisten linearen Strukturen durch mehrere Liniensegmente überdeckt. Da deren Ausrichtungen sehr ähnlich sind, können wir dies mit der lokalen Ausrichtung auch nicht vermeiden. Nach der Segmentierung der Geraden werden deshalb die überflüssigen Segmenten gelöscht.



**Abbildung 5.20:** Ein Ausschnitt der Linienmodelle von einem Museum in Heraklion direkt nach der Segmentierung der Geradenmodelle; Die Modelle wurden (a) mit bzw. (b) ohne Berücksichtigung der lokalen Ausrichtung bei der Berechnung der Erwartungen im E-Schritt erstellt.

### 5.4 Vergleich der beiden Verfahren

Kommen wir nun zu den Vor- und Nachteilen des heuristischen Verfahrens und des EM-Verfahrens. Die Eigenschaften, die in dieser Arbeit ausführlich erläutert wurden, werden in diesem Abschnitt kurz zusammengefasst und gegenübergestellt.

Ein großer Vorteil beim heuristischen Verfahren ist die schrittweise Extraktion von Geraden mit direkt anschließender Segmentierung. Wir erhalten in jedem Iterations-

schritt ein Liniensegment, das ein Hindernis der Umgebung beschreibt. Wird das Verfahren unter Umständen zu früh abgebrochen, dann vermindert dies keinesfalls die Qualität der bisher erstellten Liniensegmente. Beim EM-Verfahren hingegen können wir erst nach Beendigung der Optimierung des Geradenmodells mit dem Übergang zu Liniensegmenten beginnen. Dabei werden alle Geraden sozusagen gleichzeitig segmentiert. Erst wenn die optimale Anzahl der Geraden für das Geradenmodell gefunden wurde, kann mit einem akzeptablen Linienmodell gerechnet werden.

Bei der Anpassung der Geraden an die Messpunkte verfolgen beide Verfahren die gleiche Idee des *line-fittings*. In heuristischen Verfahren wird die Lage eines einzelnen Liniensegmentes an eine zuvor zugeordnete Punktemenge angepasst. Dabei wird das Quadrat der Summe der Abstände der Punkte zur Geraden minimiert. Dies entspricht auch dem Grundgedanken der Optimierung des Geradenmodells im EM-Verfahrens. Dort wird die Lage der Geraden an das Punktemodell angepasst, um die Wahrscheinlichkeitsfunktion zu maximieren, welche von den Abständen der Punkte zu den Geraden abhängt. Der große Unterschied zum heuristischen Verfahren ist die gleichzeitige Anpassung aller Geraden an die gesamte Menge von Messpunkten. Grundsätzlich ist die Idee der Berücksichtigung aller Messpunkte bei der Anpassung der Geraden als vorteilhaft anzusehen, aber dies erschwert natürlich auch die praktische Umsetzung. Dafür wird, im Gegensatz zum heuristischen Verfahren, eine frühzeitige Zuordnung der Messpunkte zu einer Geraden vermieden.

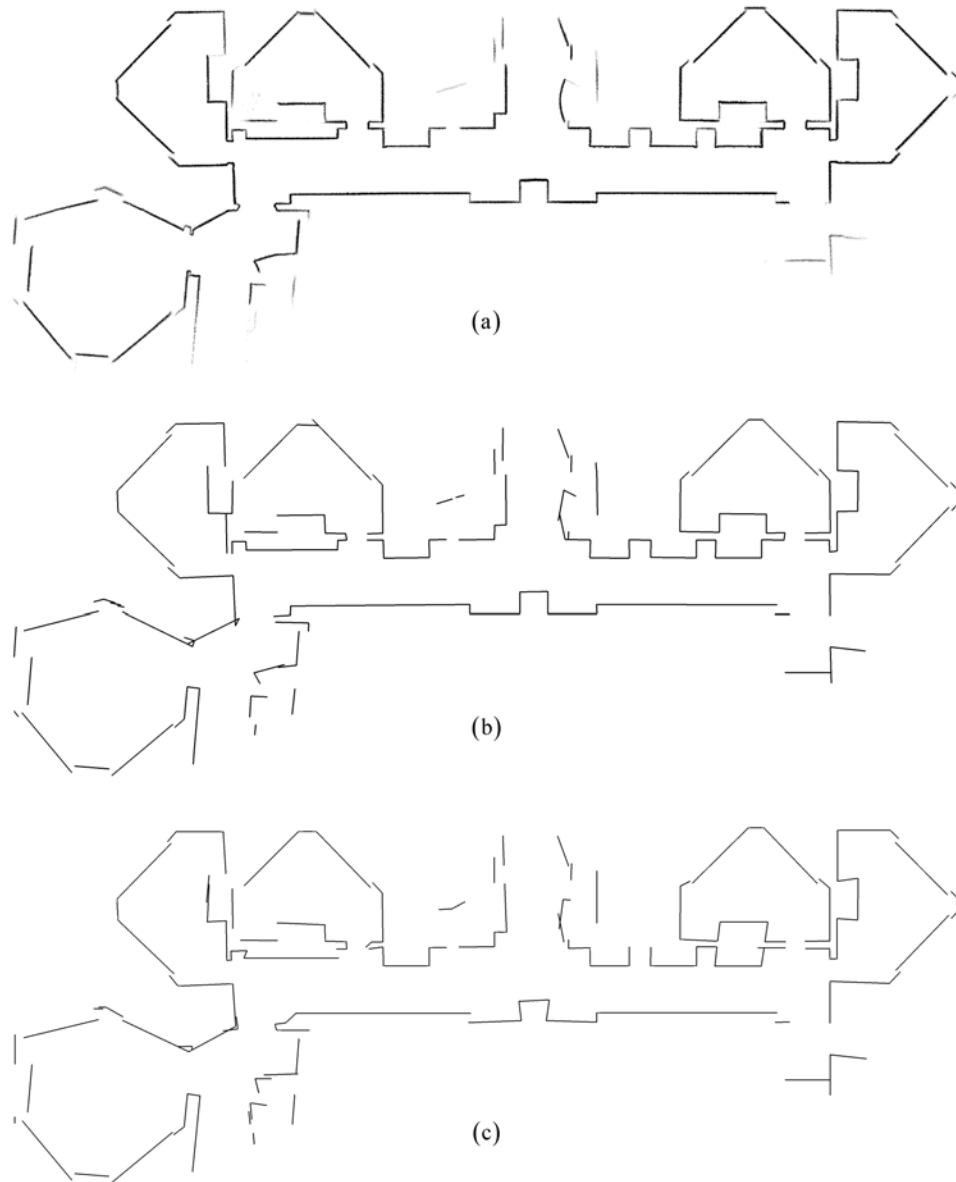
Aus den verschiedenen Ansätzen der Optimierung der Lage der Geraden ergeben sich die sehr unterschiedlichen Laufzeiten. Zum einen bestimmt die Extraktion von Geraden aus dem Hough-Raum die Laufzeit des heuristischen Verfahrens. Für das Finden des Maximums einer  $c \times r$  Akkumulatormatrix ergibt sich eine Laufzeit von  $O(c \cdot r)$ , wobei die Größe der Matrix von der Genauigkeit der Diskretisierung und vor allem von der Größe der zu vermessenden Umgebung abhängt. Zum anderen beeinflusst die Anzahl der Messpunkte die Laufzeit des Verfahrens. Für jedes erstellte Liniensegment müssen alle übrig gebliebenen Messpunkte betrachtet werden, da wir einer extrahierten Geraden alle nicht zu weit entfernten Punkte zuordnen. Insgesamt ergibt sich für das heuristische Verfahren im schlechtesten Fall eine Laufzeit von  $O(m \cdot (n + c \cdot r))$ , wobei  $n$  die Anzahl der Messpunkte und  $m$  die Anzahl der erstellten Liniensegmente darstellt. In unseren Versuchen ist die Größe des Wertes von  $c \cdot r$  ungefähr gleich der Anzahl der verwendeten Messpunkten. Für einen besseren Vergleich mit dem anderen Verfahren vereinfachen wir die Laufzeit des heuristischen Verfahrens auf  $O(n \cdot m)$ .

Die Laufzeit des EM-Verfahrens hängt vor allem von der Optimierung des Geradenmodells ab. In jedem Iterationsschritt müssen für alle Messpunkte im E-Schritt die Erwartungen neu berechnet werden, um anschließend im M-Schritt das neue Geradenmodell zu berechnen. Für ein Iterationsschritt ergibt sich eine Laufzeit von  $O(n \cdot m)$ , wobei  $n$  die Anzahl der Messpunkte und  $m$  die Anzahl der bisherigen

Geraden im Modell darstellt. Nachdem ein Geradenmodell bei der Optimierung konvergiert, wird der Nutzen jeder einzelnen Geraden überprüft. Für die Berechnung des Nutzens einer Geraden  $\theta_j$  wird im schlechtesten Fall eine Zeit von  $O(n \cdot m)$  benötigt, da sich alle minimalen Abstände der Messpunkte zu einer Geraden in dem Modell ohne  $\theta_j$  verändern können und somit neu berechnet werden müssen. Betrachten wir jedoch die Berechnungen der Nutzen zusammen, dann muss der minimale Abstand jedes Punktes insgesamt nur einmal neu berechnet werden. Daraus ergibt sich eine Laufzeit von  $O(n \cdot m)$  für die Berechnung der Nutzen der  $m$  Geraden des jeweiligen Modells. Insgesamt ergibt sich für das EM-Verfahren eine Laufzeit von  $O(i \cdot n \cdot m)$ , wobei  $i$  die Anzahl der benötigten Iterationen,  $n$  die Anzahl der Messpunkte und  $m$  die Anzahl der erstellten Geraden darstellt. Diese Laufzeit kann durch  $O(n \cdot m^2)$  nach unten beschränkt werden.

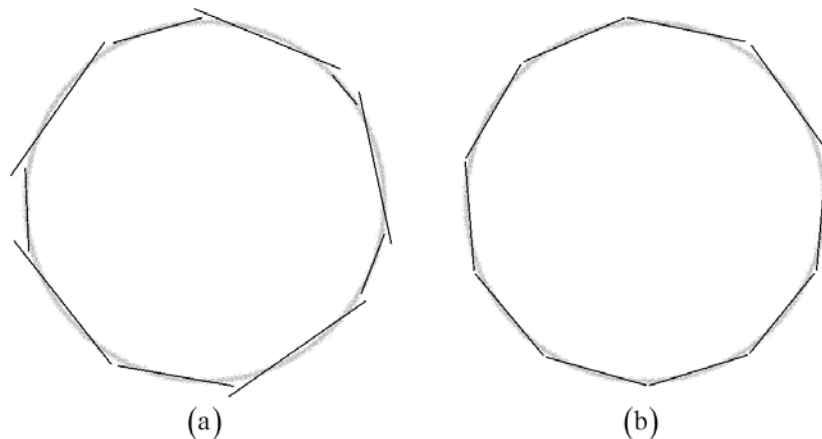
Beide Laufzeiten sind von der Anzahl der erstellten Liniensegmente bzw. Geraden abhängig. Durch den quadratischen Term in der Laufzeit des EM-Verfahrens ergibt sich jedoch eine wesentlich längere Laufzeit dieses Verfahrens, im Vergleich zum heuristischen Verfahren. Zusätzlich ist in der Praxis beim heuristischen Verfahren zu beobachten, dass im Verlauf des Verfahrens die Erstellung von Liniensegmenten immer schneller wird, da sich die zu berücksichtigende Punktemenge verringert. Beim EM-Verfahren hingegen bleibt die Punktemenge unverändert und durch die wachsende Anzahl von Geraden nimmt die Laufzeit für eine Iteration im Verlauf des Verfahrens zu. Gerade bei großen Umgebungen, für die sehr viele Geraden zur Beschreibung benötigt werden, ist das EM-Verfahren sehr langsam und kann nur mit großem Zeitaufwand durchgeführt werden.

Mit Hilfe der zahlreichen Abbildungen und Tabellen in den Abschnitten zuvor sind beim Vergleich der Ergebnisse der beiden Verfahren die Linienmodelle des heuristischen Verfahrens besser einzuschätzen. Betrachten wir für einen genaueren Vergleich die Linienmodelle in *Abbildung 5.21*. Im Modell (b) wurden die „kurzen“ Liniensegmente (vgl. *Abschnitt 3.5.3*) des heuristischen Verfahrens weggelassen, um einen besseren Vergleich mit dem Modell (c) des EM-Verfahrens zu erreichen. Erstaunlich ist die fast identische Anzahl der Liniensegmente in beiden Modellen. Dies spricht für eine gute Komplexität beider erstellten Linienmodelle und lässt auf ähnliche Modelle schließen. Schauen wir uns die Lage der Liniensegmente genauer an, dann ist die Qualität der Segmente im heuristischen Verfahren etwas besser einzustufen, da fast alle Segmente entsprechend den Hindernissen ausgerichtet sind. Im Modell (c) dagegen liegen die zumeist kürzeren Liniensegmente häufig nicht parallel zu den linearen Strukturen der Umgebung. In unseren Experimenten mit verschiedenen Umgebungen hat das heuristische Verfahren überwiegend zu besseren Ergebnissen geführt.



**Abbildung 5.21:** Modelle von einem Museum in Heraklion ( $38\text{ m} \times 18\text{ m}$ ); Das Punktemodell (a) mit 311.823 Messpunkten bildet die Grundlage der beiden anderen Modelle. Mit dem heuristischen Verfahren wurde das Linienmodell in (b) mit 140 Liniensegmenten erzeugt, ohne Einfügen der „kurzen“ Segmente. Das Linienmodell in (c) mit 139 Segmenten wurde mit dem EM-Verfahren gebildet.

In den von uns durchgeführten Experimenten wurden räumliche Umgebungen verwendet, die zumeist nur aus linearen Strukturen bestanden. Um beide Verfahren in ihrer Anwendbarkeit auf nicht-lineare Strukturen zu vergleichen, betrachten wir *Abbildung 5.22*. In diesem Beispiel wurde das simulierte Punktemodell eines Kreises mit beiden Verfahren in Linienmodelle mit jeweils 10 Liniensegmenten umgewandelt. Es ist gut zu erkennen, dass das Linienmodell *(b)* den Kreis regelmäßig und durch gleich lange Liniensegmente beschreibt. Dies ist das Resultat der gleichzeitigen Anpassung aller Geraden an das Punktemodell. Die Liniensegmente des Linienmodells *(a)* hingegen werden nacheinander extrahiert und getrennt voneinander an die Punkte angepasst, wodurch die Unregelmäßigkeiten und die unterschiedlichen Längen der Segmente entstehen. Aus den unterschiedlichen Anpassungsweisen folgte der bessere Wert des Gütemaßes  $G(\theta, Z)$  bei *(b)*. In solchen Situationen besitzt das EM-Verfahren klare Vorteile gegenüber dem heuristischen Verfahren, da zum einen alle Geraden gleichzeitig angepasst und zum anderen alle Punkte anschließend auf die Geraden für die Segmentierung verteilt werden.



**Abbildung 5.22:** Die Linienmodelle wurden mit (a) dem heuristischen Verfahren und (b) dem EM-Verfahren erstellt. Der Wert des Gütemaßes  $G(\theta, Z)$  ist bei (a) 62.006 und bei (b) 54.625, wobei beide Linienmodelle aus 10 Liniensegmenten bestehen. Das zu beschreibende simulierte Punktemodell eines Kreises mit 14.400 Messpunkten ist im Hintergrund zu sehen.

Abschließend vergleichen wir noch kurz die Ausbaufähigkeit der Verfahren. Es gibt viele Möglichkeiten, die zahlreichen Ansätze und Ideen dieser Arbeit zu verändern und teilweise könnten sogar ganze Abschnitte und (Teil-)Algorithmen ersetzt werden. Wollen wir unsere Verfahren auf 3D-Modelle, die aus Ebenen bestehen, erweitern, dann können wir die Grundideen übernehmen und an die neue Aufgabe anpassen. Beim heuristischen Verfahren muss jedoch die Hough-Transformation durch ein alternatives



Verfahren ersetzt werden, da diese Transformation in 3D nicht anwendbar ist. Das EM-Verfahren hingegen muss nicht grundlegend verändert werden. Anstatt Geraden müssen Ebenen verwendet und dementsprechend die einzelnen Verfahrensschritte verändert werden. Martin und Thrun [2002] sowie Burgard et al. [2001] haben dies bereits erfolgreich durchgeführt und können mittels EM 3D-Modelle von Umgebungen erstellen. Somit ist das EM-Verfahren im Hinblick auf die Zukunft, in der 3D-Modelle eine immer wichtigere Rolle spielen werden, das Verfahren mit dem größeren Potential.



## 6 Zusammenfassung

In dieser Arbeit wurden zwei Verfahren zur Erstellung von Linienmodellen aus Laserscannerdaten vorgestellt, welche in Zusammenarbeit mit der Arbeitsgruppe Autonome Intelligente Systeme von Prof. Dr. Wolfram Burgard entwickelt und in die Praxis umgesetzt wurden. Im Gegensatz zu anderen gängigen Verfahren, in denen Merkmale bereits aus einzelnen Scans extrahiert und mit anderen vereinigt werden, haben wir zuerst ein Punktemodell einer kompletten Umgebung erstellt und wandelten dieses Modell anschließend in ein Linienmodell um. Bei der Erstellung von Punktemodellen haben wir auf bewährte Verfahren zurückgegriffen und erhielten Modelle, in denen die lineare Strukturen der Umgebungen vollständig und deutlich zu erkennen waren.

Bevor mit der eigentlichen Erstellung eines Linienmodells begonnen wurde, haben wir verschiedene Ansätze zur Vorverarbeitung der Scandaten angewendet. Zum einen haben wir einen großen Teil von Fehlmessungen und von Messpunkten dynamischer Hindernisse herausgefiltert; zum anderen hat jeder Messpunkt eine zusätzliche Information über die Umgebung erhalten, indem wir die neu eingeführte lokale Ausrichtung zugeordnet haben. Diese konnte mit Hilfe der Extraktion von Liniensegmenten aus einem Scan gewonnen und an verschiedenen Stellen in den Verfahren eingesetzt werden. Die lokale Ausrichtung stellt die wesentliche Neuerung in unseren zwei Verfahren dar.

Wir beschäftigten uns mit verschiedenen Methoden der Linienextraktion, mit denen aus einer Menge von Messpunkten lineare Strukturen einer Umgebung erfasst werden sollten. Für die Berechnung der lokalen Ausrichtung verwendeten wir einen *split-and-merge*-Algorithmus, der auf einen einzelnen Scan angewendet wurde. Das heuristische Verfahren basiert auf der Hough-Transformation zur Extraktion von Geraden und wurde durch die lokale Ausrichtung vorteilhaft verändert. Das EM-Verfahren hingegen erstellt ein Modell mit der Methode der Erwartungsmaximierung, wobei die lokale Ausrichtung für ein akzeptables Ergebnis unverzichtbar ist. In beiden Verfahren kombinierten wir durch die Verwendung der lokalen Ausrichtung die Extraktion von Geraden aus einem einzelnen Scan mit der Extraktion von Geraden aus allen Scans.

Wie in mehreren praxisorientierten Experimenten von Innenraumumgebungen gezeigt, liefern beide von uns erstellten Verfahren kompakte und genaue Linienmodelle. Beim

Vergleich der Ergebnisse beider Verfahren sind die Linienmodelle des heuristischen Verfahrens etwas besser einzustufen. Unserer Ansicht nach besitzt das EM-Verfahren jedoch das größere Potential für die Zukunft.

### 6.1 Ausblick

Abschließend möchten wir noch einige weiterführende Ideen für unsere Verfahren vorstellen, die sich für mögliche Verbesserungen oder für Erweiterungen unserer Ansätze anbieten.

Wir beschränken uns in dieser Arbeit bei der Erstellung der Linienmodelle auf Liniensegmente, wobei die einzelnen Elemente nicht weiter spezifiziert werden. Vor allem bei Innenraumumgebungen existieren viele Ecken, die relativ einfach erkannt werden können (vgl. [Baltzakis und Trahanias 2002], [Gutmann 2000] und [Tomatis et al. 2001]). Wir könnten somit unsere Modelle um eine weitere Klasse von Merkmalen erweitern. Die meisten Ecken von Gebäuden besitzen einen  $90^\circ$  Winkel und durch diesen Umstand könnte auch die Lage der Liniensegmente verbessert werden. Eine weitere Klasse von Merkmalen stellen Zugänge und Türen dar, die sehr wichtig für die Navigation von Robotern in Umgebungen mit vielen Räumen sind. Das Auffinden dieser Merkmale wurde bereits unter anderem von Gutmann et al. [2002], Chown [2000] und Youngblood et al. [2000] vorgestellt. Unsere Verfahren könnten an bestimmten Stellen durch diese Ansätze ergänzt und unserer Modelle um einen semantischen Inhalt erweitert werden.

Im EM-Verfahren erwies sich die Verwendung von Geraden während der Erwartungsmaximierung in manchen Situationen als nachteilig. Um diese Schwäche des Verfahrens zu vermeiden, ist eine direkte Verwendung von Liniensegmenten eine theoretische Lösung. Dadurch würde jedoch der Algorithmus an einigen Stellen komplexer, da anstelle von unendlich langen Geraden Liniensegmente verwendet werden müssten, deren Länge durch die Optimierung zu bestimmen wäre. Ob es möglich ist, diese theoretische Idee in die Praxis umzusetzen, muss allerdings noch geprüft werden. Für eine praktische Umsetzung könnte zum Beispiel ähnlich der lokalen Ausrichtung den Messpunkten eine „lokale Länge“ zugeordnet werden, welche die Länge eines aus einem Scan extrahierten Liniensegment angäbe. Diese zusätzliche Information könnte bei der Bestimmung der Länge eines Liniensegments verwendet werden.

Die 3D-Umgebungsmodellierung aus Sensordaten wird in Zukunft eine immer wichtigere Rolle spielen, da 2D-Modelle nur den Umriss von Hindernissen beschreiben können und somit nur begrenzt Informationen über die Umgebung beinhalten. Da die Oberfläche von 3D-Modelle zumeist aus Polygonen besteht, ist die Bestimmung von

Ebenen ein wichtiger Bestandteil bei der Erstellung dieser Modelle. Einzelne Grundideen von Algorithmen der Merkmalsextraktion aus 2D-Messdaten könnten dafür übernommen und für diese Situation erweitert werden.

Ein Ziel der Arbeitsgruppe von Prof. Dr. Wolfram Burgard für die nahe Zukunft ist die Verwendung der lokalen Ausrichtung in bereits erstellten Verfahren der 3D-Modellierung. Das Verfahren von Hähnel et al. [2001] zur Erstellung von kompakten 3D-Modellen soll durch die lokale Ausrichtung erweitert und verbessert werden. Wie die lokale Ausrichtung in einer 3D-Umgebung definiert werden muss und ob diese Idee überhaupt angewendet werden kann, ist noch zu erforschen.



# A Parameter

Auflistung und kurze Beschreibung aller benötigten Parameter für die in dieser Arbeit vorgestellten Verfahren:

## A.1 Vermessung der Umwelt

### ***SPLIT\_DIST\_MAX***

Maximaler Abstand eines Punktes einer Menge zu einer Geraden, damit jedem Punkt der Menge diese Gerade und somit die gleiche lokale Ausrichtung zugeordnet wird (*Abschnitt 2.1.2*)

### ***SCAN\_DIST\_MAX***

Maximaler Abstand eines Messwertes zum Roboter bei der Filterung von Scandaten, damit dieser Wert mit in das Punktemodell aufgenommen wird (*Abschnitt 2.1.3*)

### ***SPLIT\_POINTS\_MIN***

Minimale Anzahl von Punkten für die Bildung einer Optimalen Geraden in der *split&merge*-Funktion, damit diese Punkte mit in das Punktemodell aufgenommen werden (*Abschnitt 2.1.3*)

### ***DELETE\_GRID\_MIN***

Minimale Reflektionswahrscheinlichkeit einer Rasterzelle bei der Filterung von dynamischen Messpunkten, damit die Messpunkte in dieser Zelle nicht aus dem Punktemodell entfernt werden (*Abschnitt 2.2.3*)

## A.2 Heuristisches Verfahren

### ***ASSIGN\_DIST\_MAX***

Maximaler senkrechter Abstand eines Messpunktes zu einer Geraden, damit der Punkt bei der Segmentierung der Geraden zugeordnet wird (*Abschnitt 3.2.1*)

### ***SEG\_DIST\_MAX***

Maximaler Abstand zweier aufeinander folgender Punkte bei der Segmentierung einer Geraden, damit beide Punkte dem gleichen Hindernis bzw. dem gleichen Liniensegment zugeordnet werden (*Abschnitt 3.2.2*)

### ***SEG\_LENGTH\_MIN***

Minimale Länge eines zulässigen Liniensegments bei der Filterung der Liniensegmente einer Geraden (*Abschnitt 3.2.3*)

### ***AKKU\_MIN***

Abbruchkriterium des Verfahrens; Ist das Maximum der Akkumulatorzellen bei der Hough-Transformation kleiner *AKKU\_MIN*, dann wird keine neue Gerade mehr aus dem Hough-Raum extrahiert (*Abschnitt 3.1.4* und *3.4*).

### ***MERGE\_OVERLAP\_MIN***

Minimale Überlappung zweier Geraden für eine zulässige Vereinigung, unter Berücksichtigung von *MERGE\_DIST\_MAX* (*Abschnitt 3.5.1*)

### ***MERGE\_DIST\_MAX***

Maximaler Abstand beider Endpunkte eines Liniensegments zu einem anderen, damit beide Liniensegmente vereinigt werden, unter Berücksichtigung von *MERGE\_OVERLAP\_MIN* (*Abschnitt 3.5.1*)

### ***EXTEND\_LENGTH\_MAX***

Maximale Länge einer Verlängerung bzw. Verkürzung eines Liniensegments, damit anschließend ein Endpunkt auf einem anderen Segment liegt (*Abschnitt 3.5.2*)

### ***SHORT\_SEGMENT\_MAX***

Maximale Länge eines „kurzen“ Liniensegments, das für die Verbindung zweier Liniensegmente erstellt wird (*Abschnitt 3.5.3*)



## A.3 EM-Verfahren

$\sigma$

Varianz des Abstandes eines Punktes zu einer Geraden; Diese wird bei der Berechnung der Erwartungen  $E[c_{ij}|\theta, Z]$ , dass ein Punkt zu einer Geraden zugeordnet wird, benötigt (*Abschnitt 4.1 - 4.3*).

$\sigma^*$

Varianz der Differenz zwischen der lokalen Ausrichtung eines Punktes und der Ausrichtung einer Geraden; Diese wird beim erweiterten Ansatz der Berechnung der Erwartungen  $E^*[c_{ij}|\theta, Z]$  benötigt (*Abschnitt 4.3.1*).

### ***PENALTY***

Gewichtung des Strafterms bei der Berechnung des Gütemaßes eines Geradenmodells bzw. des Nutzens einer Geraden (*Abschnitt 4.4*); Die Kosten für eine einzelne Gerade sind gleich dem Wert von *PENALTY*.

### ***NEW\_DIST\_MIN***

Ist der kürzeste Abstand eines Punktes zu einer Geraden in einem Modell größer als *NEW\_DIST\_MIN*, dann wird dieser Punkt beim Einfügen einer neuen Gerade mit berücksichtigt. Hat jeder Messpunkt einen kleineren Abstand als *NEW\_DIST\_MIN* zu einer Geraden, dann wird keine neue Gerade mehr eingefügt (*Abschnitt 4.4 und 4.5*).

### ***ERROR\_MIN***

Minimale Anzahl von hintereinander eingefügten Geraden die das Gütemaß nicht verbessern, damit die Suche nach einem optimalen Geradenmodell bzw. nach der optimale Anzahl von Geraden für ein Modell beendet wird (*Abschnitt 4.5*)

### ***SEG\_DIST\_MAX***

Maximaler Abstand zweier aufeinander folgender Punkte bei der Segmentierung einer Geraden, damit beide Punkte dem gleichen Hindernis bzw. dem gleichen Liniensegment zugeordnet werden (*Abschnitt 4.6*)

### ***SEG\_LENGTH\_MIN***

Minimale Länge eines zulässigen Liniensegments bei der Filterung von Liniensegmenten einer Geraden (*Abschnitt 4.6*)

### ***DELETE\_DIST\_MAX***

Maximale Abstände beider Endpunkte eines Liniensegments zu einem anderen, damit das Liniensegment gelöscht werden kann, wenn beide Endpunkte überlappen (*Abschnitt 4.7.1*)

***EXTEND\_LENGTH\_MAX***

Maximale Länge einer Verlängerung bzw. Verkürzung eines Liniensegments, damit anschließend der Endpunkt auf einem anderen Segment liegt (*Abschnitt 4.7.2*)

# Literaturverzeichnis

- [Arras und Siegwart 1997] Arras, K.O. und Siegwart, R.Y. (1997). *Feature Extraction and Scene Interpretation for Map-Based Navigation and Map Building*. In: *Proc. SPIE, Mobile Robotics XII*, volume 3210.
- [Arras et al. 1996] Arras, K.O., S.J. Vestli und N.N. Tschichold-Gürman (1996). *Echtzeitfähige Merkmalsextraktion und Situationsinterpretation aus Laserscannerdaten*. In: *Autonome Mobile Systeme (AMS'96)*, Seiten 57-66, München, Springer-Verlag.
- [Baltzakis und Trahanias 2002] Baltzakis, H. und Trahanias, P. (2002). *An Iterative Approach for Building Feature Maps in Cyclic Environments*. In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02)*.
- [Bennewitz et al. 2002] Bennewitz, M., W. Burgard und S. Thrun (2002). *Using EM to Learn Motion Behaviours of Persons with Mobile Robots*. In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02)*.
- [Burgard et al. 1998] Burgard, W., A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D.Schulz, W. Steiner und S. Thrun (1998). *The interactive museum tour-guide robot*. In: *Proc. of the National Conference on Artificial Intelligence (AAAI'98)*.
- [Burgard et al. 2001] Burgard, W., Y. Liu, R. Emery, D. Chakrabarti und S. Thrun (2001). *Using EM to Learn 3D Models of Indoor Environments with Mobile Robots*. In: *Proc. of the Eighteenth International Conference on Machine Learning*.
- [Chown 2000] Chown, E. (2000). *Gateways: An approach to parsing spatial domains*. In: *Workshop on Machine Learning of Spatial Knowledge (ICML'00)*.

- [Crowley 1989] Crowley, J.L. (1989). *World Modelling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging*. In: *Proc. of the International Conference on Robotics and Automation (ICRA'89)*.
- [Douglas und Peucker 1973] Douglas, D.H. und Peucker, T.K. (1973). *Algorithms for the reduction of the number of points required to represent a line or its caricature*. The Canadian Cartographer.
- [Duda und Hart 1973] Duda, R. und Hart, P. (1973). *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York.
- [Gonzalez et al. 1994] Gonzalez, J., A. Ollero und A. Reina (1994). *Map Building for a Mobile Robot equipped with a 2D Laser Rangefinder*. In: *Proc. of the International Conference on Robotics and Automation (ICRA'94)*.
- [Grandjean und de Saint Vincent 1989] Grandjean, P. und de Saint Vincent, A.R. (1989). *3-D Modelling of Indoor Scenes by Fusion of Noisy Range and Stereo Data*. In: *Proc. of the International Conference on Robotics and Automation (ICRA'89)*.
- [Gutmann 2000] Gutmann, J.S. (2000). *Robuste Navigation autonomer mobiler Roboter*. Dissertation, Institut für Informatik, Universität Freiburg.
- [Gutmann et al. 2002] Gutmann, J.S., D. Schröter und M. Beetz (2002). *RG Mapping: Learning Compact and Structured 2D Line Maps of Indoor Environments*. In: *Proc. of the International Workshop on Robot and Human Interactive Communication (ROMAN'02)*, Berlin, Germany.
- [Haberäcker 1995] Haberäcker, P. (1995). *Praxis der digitalen Bildverarbeitung und Mustererkennung*. Carl Hanser Verlag München Wien.
- [Hähnel et al. 2001] Hähnel, D., W. Burgard und S. Thrun (2001). *Learning compact 3D models of indoor and outdoor environments with a mobile robot*. In: *The fourth European workshop on advanced mobile robots (EUROBOT'01)*.
- [Hähnel et al. 2003] Hähnel, D., W. Burgard, D. Fox und S. Thrun (2003). *A efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements*. In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03)*.

- [Hershberger und Snoeyink 1992] Hershberger, J. und Snoeyink, J. (1992). *Speeding Up the Douglas-Peucker Line-Simplification Algorithm*. In: *Proc. of the 5<sup>th</sup> International Symposium Spatial Data Handling*, Seiten 134-143.
- [Hough 1962] Hough, P.V.C. (1962). *Method and Means for recognising complex patterns*. US Patent 3069654.
- [Krengel 1991] Krengel, U. (1991). *Einführung in die Wahrscheinlichkeitstheorie und Statistik*. Vieweg Verlag.
- [Leonard et al. 2001] Leonard, J.J., P.M. Newman, R.J. Rikoski, J.N. Neira und J.D. Tardós (2001). *Towards Robust Data Association and Feature Modeling for Concurrent Mapping and Localization*. In: *Proc. of the Tenth International Symposium on Robotics Research*, Lorne, Victoria, Australia.
- [Lu und Milios 1994] Lu, F. und Milios, E.E. (1994). *Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans*. In: *Proc. of the Computer Vision and Pattern Recognition Conference (CVPR'94)*.
- [Martin und Thrun 2002] Martin, C. und Thrun, S. (2002). *Real-Time Acquisition of Compact Volumetric Maps with Mobile Robots*. In: *Proc. of the International Conference on Robotics and Automation (ICRA'02)*.
- [Moravec und Elfes 1985] Moravec, H.P. und Elfes, A. (1985). *High Resolution Maps from Wide Angle Sonar*. In: *Proc. of the International Conference on Robotics and Automation (ICRA'85)*.
- [Mitchell 1997] Mitchell, T.M. (1997). *Machine Learning*. McGraw-Hill.
- [Newman et al. 2002] Newman, P.M., J.J. Leonard, J.N. Neira und J.D. Tardós (2002). *Explore and Return. Experimental Validation of Real-Time Concurrent Mapping and Localization*. In: *Proc. of the International Conference on Robotics and Automation (ICRA'02)*.
- [Pitas 2000] Pitas, I. (2000). *Digital Image Proceeding: Algorithms and Applications*. John Wiley & Sons.

- [Press et al. 1992] Press, W.H., S.A. Teukolsky, W.T. Vetterling und B.P. Flannery (1992). *Numerical Recipes in C: The art of scientific computing*. Cambridge University Press, Cambridge, 2<sup>nd</sup> edition.
- [Pfister 2002] Pfister, S.T. (2002). *Weighted Line Fitting and Merging*. Tech. Rep., California Institute of Technology.
- [Pfister et al. 2003] Pfister, S.T., S.I. Roumeliotis und J.W. Burdick (2003). *Weighted Line Fitting Algorithms for Mobile Robot Map Building and Efficient Data Representation*. In: *Proc. of the International Conference on Robotics and Automation (ICRA'03)*.
- [Russel und Norvig 1994] Russel, S. und Norvig, P. (1994). *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ.
- [Tomatis et al. 2001] Tomatis, N., I. Nourbakhsh und R. Siegwart (2001). *Simultaneous Localization and Map Building: A Global Topological Model with Local Metric Maps*. In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01)*.
- [Thrun et al. 2003] Thrun, S., D. Hähnel, R. Triebel und W. Burgard (2003). *Map Building with Mobile Robots in Dynamic Environments*. In: *Proc. of the International Conference on Robotics and Automation (ICRA'03)*.
- [Wang und Thorpe 2002] Wang, C.-C. und Thorpe, C. (2002). *Simultaneous Localization and Mapping with Detecting and Tracking of Moving Objects*. In: *Proc. of the International Conference on Robotics and Automation (ICRA'02)*.
- [Youngblood et al. 2000] Youngblood, G.M., L.B. Holder und D.J. Cook (2000). *A Framework for Autonomous Mobile Robot Exploration and Map Learning through the use of Place-Centric Occupancy Grids*. In: *Proc. of the Machine Learning Workshop on Learning From Spartial Information, 2000*.