

Implementation und Evaluation verschiedener Scanmatcher für das mobile System Robbie

Studienarbeit im Studiengang Computervisualistik

vorgelegt von

Susanne Maur

Betreuer: Prof. Dr.-Ing. Dietrich Paulus, Institut für Computervisualistik,
Fachbereich Informatik

Koblenz, im Februar 2009

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Richtlinien der Arbeitsgruppe für Studien- und Diplomarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden. ja nein

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu. ja nein

Koblenz, den

Unterschrift

Inhaltsverzeichnis

1	Einleitung	7
2	Stand der Wissenschaft	9
2.1	Iterative Closest Points	10
2.2	Iterative Dual Correspondences	14
2.3	Metric Based Iterative Closest Points	18
3	Eigener Ansatz	29
3.1	Implementation	29
3.1.1	Konvergenzkriterien	29
3.1.2	Fehlerfall	30
3.1.3	Klassenstruktur	31
3.1.4	Ablauf Scanmatching	33
3.1.5	Entfernen falscher Punkt Korrespondenzen	34
3.2	Vorverarbeitung der Laserdaten	34
3.2.1	Entfernen der Ausreißer	34
3.2.2	Löschen ungültiger Punkte	36
3.2.3	Entfernen kleiner Punkt-Cluster	36

3.2.4	Extraktion von Linien	38
3.2.5	Projektions- und Sichtbarkeitsfilter	40
3.2.6	Reduzierung der Punkte	44
3.3	Einbindung in das mobile System Robbie	46
3.3.1	Verwendete Daten	46
3.3.2	Koordinaten Systeme	47
3.3.3	Einbindung in Kartenerstellung und Selbstlokalisierung	50
4	Experimente und Ergebnisse	59
4.1	Parameterjustierung	64
4.1.1	Konvergenzkriterien	64
4.1.2	Metrischer Faktor	68
4.1.3	Eliminierung falscher Punkt-Korrespondenzen	70
4.1.4	Maskengröße des Reduktionsfilters	75
4.1.5	Maximale Distanz der Korrespondenzpunkte	77
4.1.6	Maximale Winkeldistanz der Korrespondenzpunkte	79
4.2	Vergleich der Verfahren	81
4.3	Integration in Partikelfilter	83
5	Zusammenfassung	91
6	Ausblick	93
	Literaturverzeichnis	96

Kapitel 1

Einleitung

Motivation Eines der Hauptprobleme autonomer mobiler Systeme betrifft die Lokalisation und Kartenerstellung. Diese beiden Probleme sind besonders schwierig, da sie sich gegenseitig bedingen. Ein wichtiger Aspekt der Kartenerstellung ist es, die Bewegung zu bestimmen, welche der Roboter zwischen zwei diskreten *Mapping-Schritten* vollführt hat. Diese Bewegung wird repräsentiert durch die Transformation zwischen den aufeinanderfolgenden Roboterposen.

Um diese Transformation zu bestimmen, können verschiedene Ansätze gewählt oder auch kombiniert werden. Das mobile System *Robbie* arbeitet bislang mit einem Partikelfilter, welcher durch Odometriedaten gestützt wird. Die Odometriedaten sind jedoch häufig sehr ungenau und können durch verschiedene Faktoren negativ beeinflusst werden.

Alternativ könnte jene Transformation auch aus der Verschiebung der Laserdaten ermittelt werden. Solche Verfahren werden als Scanmatching-Verfahren bezeichnet. Ein solches Vorgehen verspricht eine wesentlich höhere Genauigkeit als durch die reine Verwendung der Odometriedaten erlangt werden kann.

Das mobile System *Robbie* nimmt bei den jährlichen RoboCup Wettbewerben teil. Sein Schwerpunkt liegt dort auf Autonomie und Kartenerstellung der Rescue Wettkämpfe. In dieser Teildisziplin hat es 2007 in Atlanta(USA) und 2008 in Suzhou(China) den Weltmeistertitel erlangt. Daher ist es von besonderer Wichtigkeit, die Entwicklung in diesen Bereichen voranzutreiben.

Verwandte Arbeiten Paul Besl und Neil McKay veröffentlichten bereits 1992 das bislang bekannteste iterative Scanmatching-Verfahren, den *Iterative Closest Points Algorithmus (ICP)* [BM92]. Feng Lu und Evangelos Miliotis zogen im Jahre 1994 mit dem *Iterative Dual Correspondences (Idc)* [LM94] gleichauf und brachten im Jahre 1997 ein geschlossenes Verfahren zur Transformationsbestimmung basierend auf Punktkorrespondenzen heraus [LM97]. Seit dieser Zeit gab es viele kleinere Verbesserungsvorschläge. So evaluierten J.-S. Gutmann und Chr. Schlegel zum Beispiel 1996 die damals gängigsten Verfahren und kombinierten sie zu einem gemeinsamen Scanmatcher [GS96]. Die nächste neuartige Veröffentlichung machten 2005 Javier Miguez, Florent Lamiroux und Luis Montesano mit ihrem *Metric based ICP (MbICP)*. Dieser beruht immernoch auf dem Grundprinzip des *ICPs*, jedoch ist er nach Tests der Autoren wesentlich stabiler, schneller und genauer als die bisherigen Scanmatching-Verfahren.

Problemstellung und Lösungsansatz Im Rahmen dieser Studienarbeit werden drei verschiedene Scanmatching-Verfahren implementiert und deren Effizienz evaluiert. Ein besonderes Augenmerk wird hierbei auf den *MbICP* gerichtet, um seine postulierte Überlegenheit in neutralen Tests zu überprüfen. Die Verfahren werden in das mobile System *Robbie* integriert und seine Kartenerstellung und Selbstlokalisierung eingebunden. Durch situationsabhängiges Verhalten soll eine möglichst gute Nutzung der verschiedenen Algorithmen erreicht werden. Desweiteren sollen die Matching-Verfahren auch anderen Anwendungen im Rahmen des *Robbie*-Systems zur Verfügung stehen.

Überblick Der Aufbau der Studienarbeit ist wie folgt: Nach einer kurzen Einführung in das Thema des Scanmatchings wird anhand der theoretischen Basis von *Icp*, *Idc* und *MbIcp* der aktuelle Stand der Technik vorgestellt. Im nächsten Kapitel folgt die Beschreibung des eigenen Ansatzes. Dieser umfasst die strukturellen Aspekte der Implementation, eigene Modifikationen und die Einbindung der Verfahren in die Kartenerstellung von *Robbie*. Im Anschluss findet sich die Evaluation der Verfahren. Dort werden Effizienztests der wichtigsten Programmparameter durchgeführt und die Wirkungsweise des Scanmatchers im Zuge der Kartenerstellung evaluiert. In letzten Kapitel folgt dann eine Zusammenfassung der Ergebnisse mit Ausblick auf weitere Nutzungs- und Forschungsbereiche.

Kapitel 2

Stand der Wissenschaft

Mit einem Matching-Verfahren versucht man, verschiedene Datensätze aufeinander abzubilden. Bei dem Problem des Scanmatchings handelt es sich bei den verwendeten Daten um Laserdaten, die in diskreten Zeitabständen voneinander aufgenommen wurden. Aus diesen lassen sich Punktwolken im zwei dimensional Raum berechnen, worauf die Matching-Verfahren angewendet werden können. Das Ziel hierbei ist es, die Verschiebung zu berechnen, die zwischen den Aufnahmeorten zweier aufeinanderfolgender Laser-Scans erfolgte. Mit anderen Worten: gesucht ist die relative Transformation, die den einen der Laserscans auf einen anderen abbildet.

Im folgenden seien drei Scanmatching Verfahren beschrieben, die diese Problemstellung angehen. In Abschnitt 2.1 wird der Algorithmus des *Iterative Closest Points* (Icp) erklärt, welcher in dieser Form gemeinsam von [BM92, Besl/McKay] und [LM94, Lu/Milios] entwickelt wurde. Die Idee wurde 1992 von [BM92, Besl/McKay] entwickelt, wohingegen [LM94, Lu/Milios] 1994 eine effizienteres Optimierungsverfahren ergänzte. Der Grundidee des *Icps* bildet ebenfalls die Basis und Ausgangssituation für die folgenden beiden Verfahren. So stellen die Algorithmen des *Iterative Dual Correspondences* (Idc) von [LM97, Lu/Milios] in Abschnitt 2.2 und des *Metric based Icp* (MbIcp) von [MLM05, Minguez et. al.] in Abschnitt 2.3 in erster Linie Verbesserungsvorschläge für den *Icp* dar. Die Änderungen beziehen sich hauptsächlich auf die Berechnung der Rotationskomponente und versprechen qualitativ bessere Ergebnisse. Auch erhofft man sich eine schnellere Termi-

nation der Berechnungen.

Bei allen drei Verfahren handelt es sich um ein iteratives Vorgehen. Es werden in jeder Iteration Korrespondenzpartner bestimmt, welche durch ein Optimierungsverfahren aufeinander abgebildet werden. Wie dies im einzelnen geschieht, ist verfahrensabhängig.

2.1 Iterative Closest Points

Idee

Als Eingabe gegeben sind uns zwei Datensätze mit Punkten in einer Ebene. Wir bezeichnen den älteren der beiden Datensätze als Referenz-Scan S_{ref} und den neueren als aktuellen Scan S_{cur} . Des weiteren kann eine ungefähre Näherung der Transformation q_0 zwischen den beiden Scans gegeben sein, die z.B. durch Odometrie-Daten oder mittels Inertial-Sensor gewonnen wurde. Ist zu Beginn des Verfahrens keine ungefähre Transformation gegeben, so initialisieren wir q_0 mit Null.

Das im folgenden vorgestellten Verfahren von [BM92] arbeiten iterativ und gehen das Problem in zwei Schritten an. In jeder Iteration k wird eine Menge C an Korrespondenzen zwischen den beiden Scans bestimmt sowie die Transformation q_k , die den quadratischen Fehler zwischen allen Korrespondenzpaaren minimiert.

Daraufhin transformiert man S_{cur} mit q_k und beginnt erneut mit der Korrespondenz- und Transformationsbestimmung. Das Ziel des Verfahrens ist es, durch Iteration q_0 solange zu verbessern, bis die optimale Transformation q zwischen den beiden Scans gefunden ist. q ergibt sich hierbei aus der Summe über alle Teiltransformationen q_i .

Eine Transformation $q = (x, y, \theta)$ beschreibt sowohl eine Verschiebung in x- und y- Richtung als auch eine Rotation θ um dem Ursprung des Koordinatensystems.

Umsetzung

Iterative Closest Points Algorithmus Dieses Verfahren wird bis zur Konvergenz wiederholt:

- Zunächst transformiert man jeden Punkt p'_i von S_{new} in das Referenzsystem von S_{ref} durch die Näherung q_k mit $c_i = q_k(p'_i)$. Nun bestimmt man für jeden Punkt c_i den nächstgelegenen Punkt p_j aus dem Referenz-Scans S_{ref} .

$$\min\{d(c_j, p_j)\} \quad (2.1)$$

Das Ergebnis ist eine Menge C mit n Korrespondenzen (p_j, c_j) .

- Dann bestimmt man die Näherung für die Transformation, die den quadratischen Fehler zwischen den Korrespondenzen aus C minimiert. Hierzu minimiert man die folgende Gleichung über q :

$$E_{dist}(q) = \sum_{i=1}^n d(r_j, q(c_j))^2 \quad (2.2)$$

Wenn das Verfahren konvergiert, so ist q_{min} die gesuchte Näherung der Transformation. Wenn nicht, so iteriert man ein weiteres mal mit $q_{k+1} = q_{min}$.

Die Kriterien zu Korrespondenzbestimmung und Minimierung des quadratischen Fehlers sind so zu wählen, dass das Verfahren stets konvergiert.

Korrespondenz: nächster Punkt Ein entscheidender Schritt des Verfahrens ist die Bestimmung der Korrespondenzen. Wie der Name des *Iterative Closest Points*-Algorithmuses vermuten lässt, wählt man hierzu jeweils den nächstgelegenen Punkt aus. Das heißt für jeden transformierten Punkt $c_i = q_k(p'_i)$ des aktuellen Scans bestimmt man den nächstgelegenen Punkt p_j des Referenz-Scans S_{ref} . Somit erhält man die gesuchte Menge C mit Punktkorrespondenzen (p_j, c_j) . Als Abstandsmaß wird die euklidische Distanz verwendet.

Transformation Punkt-Linie Der Aufwand der Scanmatching Verfahren ist im Allgemeinen abhängig von der Menge der Korrespondenzen, die bestimmt werden müssen. Die Größe der aufeinander zu matchenden Datensätze ist also sehr entscheidend für die Laufzeit der Verfahren. Es liegt daher nahe, die Datensätze zunächst zu reduzieren. Dadurch verliert man jedoch Genauigkeit bei der Bestimmung der Transformation. Um diesen Effekt zu mindern, kann man bei der Bestimmung der Korrespondenzen zwischen den Punkten

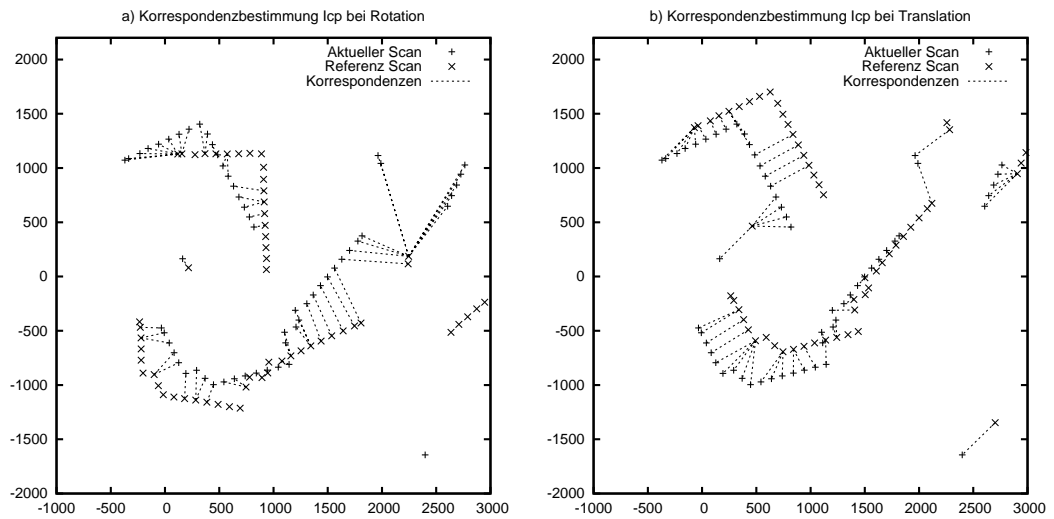


Bild 2.1: Icp: Korrespondenzzuordnungen bei Rotation und Translation. Reduktion der Scanpunkte auf 10 cm Abstände. a) Rotation um -25° Grad; b) Translation um 30 cm in x- und y-Richtung.

des Referenz-Scans interpolieren. Wir suchen also für jeden Punkt des aktuellen Scans nicht mehr nach einem korrespondierenden Punkt im Referenz-Scan, sondern einen korrespondierenden Punkt der auf einer Verbindungslinie zwischen zwei Referenz-Punkten liegt. Somit betrachten wir im folgenden stets auch die Transformation zwischen Punkt und Linie.

Dies ändert die Korrespondenz-Bestimmung des Verfahrens 2.1 wie folgt:

- Zunächst transformiert man jeden Punkt p'_i von S_{new} in das Referenzsystem von S_{ref} durch die Näherung q_k mit $c_i = q_k(p'_i)$. Da es sich um diskrete Daten handelt, kann man davon ausgehen, dass es zwischen zwei aufeinander folgenden Punkten (p_i, p_{i+1}) von S_{cur} eine lokale Struktur gibt. Daher gehört der korrespondierende Punkt p_j von c_j zu dem Liniensegment $[p_i, p_{i+1}]$:

$$\min\{d(c_j, [p_i, p_{i+1}])\} \quad (2.3)$$

Das Ergebnis ist wiederum eine Menge C mit n Korrespondenzen (p_j, c_j) .

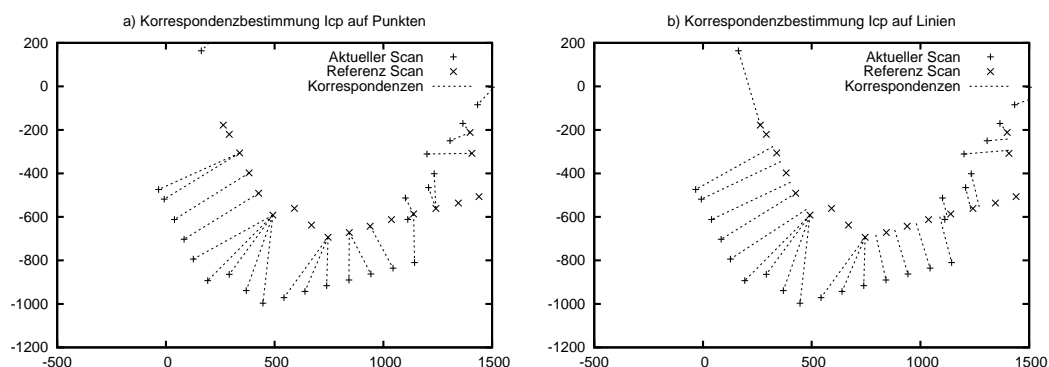


Bild 2.2: Vergleich: Zuordnungen Punkt-Punkt und Punkt-Linie. Reduktion der Scanpunkte auf 10 cm Abstände; Translation um 30 cm in x - und y -Richtung. a) Zuordnung auf der Basis von Punkten; b) Zuordnung auf der Basis von Punkt und Linie durch Interpolation der Referenzpunkte.

Als Abstandsmaß ist wieder die euklidische Distanz zu wählen.

Minimierung der kleinsten quadratischen Abstands Unter der Annahme, dass n Punkt-korrespondenzen (p_i, p'_i) vorkommen und als korrekte Scanwerte vorliegen, beschreibt das Minimum der folgenden Fehlerfunktion die gesuchte Transformation vom Referenzsystem in das aktuelle Koordinatensystem:

$$E_{dist}(q) = \sum_{i=1}^n \|R_\theta \cdot p_i + T_{xy} - p'_i\|^2, \quad (2.4)$$

wobei R_θ die Rotationsmatrix darstellt, die die Rotation um dem Winkel θ beschreibt, und T_{xy} die Translationsmatrix der x - und y -Komponente.

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, T_{xy} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Nach [LM97] existiert eine geschlossene Lösung in q zur Minimierung von $E_{dist}(q)$ bei

gegebenen Punkt Korrespondenzen, nämlich:

$$\theta = \arctan\left(\frac{S_{xy'} - S_{yx'}}{S_{xx'} + S_{yy'}}\right) \quad (2.5)$$

$$x = \bar{x}' - (\bar{x} \cdot \cos\theta - \bar{y} \cdot \sin\theta) \quad (2.6)$$

$$y = \bar{y}' - (\bar{x} \cdot \sin\theta + \bar{y} \cdot \cos\theta) \quad (2.7)$$

mit

$$\bar{p} = \frac{1}{n} \sum_{i=1}^n p_i$$

$$\bar{p}' = \frac{1}{n} \sum_{i=1}^n p'_i$$

$$S = \begin{pmatrix} S_{xx'} & S_{xy'} \\ S_{yx'} & S_{yy'} \end{pmatrix} = \sum_{i=1}^n (p_i - \bar{p})(p'_i - \bar{p}')^T$$

Hierbei stellt \bar{p} den Mittelpunkt des Referenzscans und \bar{p}' den Mittelpunkt des aktuellen Scans dar. Bei der Matrix S handelt es sich um die Kovarianzmatrix.

Zusammenfassung Das Verfahren des *Icps* konvergiert stets nach endlich vielen Schritten. Mit diesem Verfahren bestimmt man sowohl die Rotations- als auch die Translationskomponente in einem. Der *Icp* benutzt den euklidischen Abstand zur Korrespondenzbestimmung und ebenso zur Minimierung des quadratischen Fehlers.

2.2 Iterative Dual Correspondences

Idee

Die Schwächen des *Icp*-Algorithmus begründen sich darin, dass der Abstand zwischen zwei korrespondierenden Punkten bei einer Rotation proportional zunimmt, je weiter der Punkt vom Ursprung des Rotationszentrums entfernt ist. Dies führt dazu, dass bei größeren Rotationen die richtige Zuordnung der Korrespondenzen unwahrscheinlich wird. Ein weiterer Nachteil des *Icp*-Algorithmus ist, dass er relativ langsam konvergiert. Besonders

wenn die Struktur des Scans kurvig ist, enthalten die vom *Icp* gefundenen Korrespondenzen nur wenig Information über die Rotation. Der *Icp*-Algorithmus tendiert dazu, große Rotationen durch Translationen auszugleichen, was natürlich zu Fehlern führt.

Der *Iterative Dual Correspondences (Idc)* Algorithmus erweitert den *Icp*-Algorithmus daher um eine weitere Korrespondenzbestimmung. Bei dieser zweiten Korrespondenzbestimmung wird primär die Entfernung des Punktes zum Ursprung berücksichtigt. Die gesuchte Transformation q wird dann aus beiden Korrespondenzmengen ermittelt.

Umsetzung

Korrespondenz: gleiche Entfernung Die zusätzliche Korrespondenzbestimmung zielt darauf ab, hauptsächlich die Rotationskomponenten zu bestimmen.

Betrachten wir einen Datenpunkt c_i und seinen Korrespondenzpunkt p_i mit $q(c_i) = p_i$ und $q = (x, y, \theta)$. Ignorieren wir die Translation in x- und y-Richtung, so verbleibt lediglich die Rotation θ um den Ursprung des Koordinatensystems. Dann ist $|c_i| \approx |p_i|$. Desweiteren hängen die Polarwinkel ω_c von c_i und ω_p von p_i zusammen durch $\omega_p \approx +\omega_c + \theta$. Dies impliziert, dass die Korrespondenz von c_i nach einer Rotation von θ einen Punkt mit dem gleichen polaren Abstand wie c_i ergeben. Außerdem unterscheiden sich die Polarwinkel der Korrespondenzen durch den Rotationswinkel θ . Bei einer kleinen Translation können wir davon ausgehen, dass der Punkt p_i mit dem gleichen Abstand wie c_i eine gute Näherung für die echte Korrespondenz von c_i ergibt. Diese Korrespondenz liefert eine reichhaltige Information über den Rotationswinkel θ .

Um sicher zu stellen dass unser neues Verfahren nur eine verlässliche Korrespondenz findet, suchen wir nur nach einem Punkt mit dem gleichen Abstand in der lokalen Umgebung von c_i . Angenommen wir können eine Grenze B_θ bestimmen für die Rotation θ , so dass gilt: $|\theta| \leq B_\theta$. Das bedeutet, dass p_i in dem Sektor liegen sollte, der durch $\omega \pm B_\theta$ eingeschlossen wird.

Somit ergibt sich folgende *gleiche-Entfernungs-Regel*:

Für einen Scan-Punkt c_i ist seine korrespondierender Punkt p_i in der Welt, wobei gilt dass $|\omega_p - \omega_c| \leq B_\theta$ und $|p_i|$ ist am nächsten an $|c_i|$. Abbildung (..) verdeutlicht diese Regel.

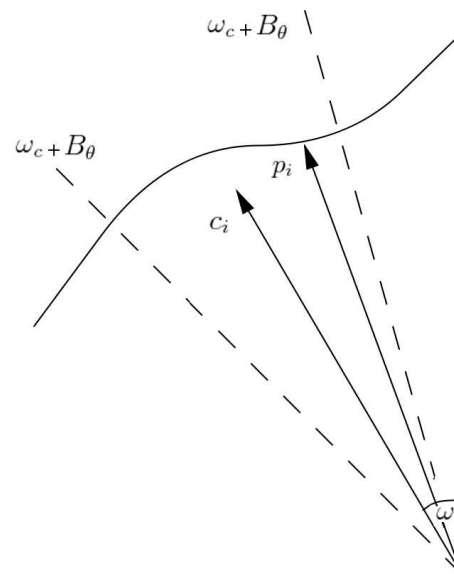


Bild 2.3: IMRP: Korrespondenzbestimmung nach der gleichen-Entfernungs-Regel. Quelle: [LM94].

Minimierung der kleinsten quadratischen Abstands Zur Berechnung der Transformation zur Minimierung des kleinsten quadratischen Abstands zwischen den Punkt-Korrespondenzen verwenden wir das gleiche Verfahren wie beim *Icp*.

Iterative Matching-Range-Points Algorithmus Lediglich auf Basis der *gleiche-Entfernungs-Regel* kann man das *Iterative Matching-Range-Points* (IMRP) Verfahren definieren, wie 1994 von [LM94] vorgestellt. Es ist hierbei zu empfehlen, den Parameter B_θ zur Bestimmung der Nachbarschaftsgröße mit jeder Iteration zu verringern. Da B_θ ebenfalls die maximal mögliche Rotation eingrenzt, erscheint es sinnvoll B_θ von der bestimmten Rotation der vergangenen Iteration abzuleiten.

Nach Versuchen von [LM94] konvergiert die Rotationskomponente mittels IMRP wesentlich schneller als mittels *Icp*. Die Translations-Komponente hingegen wird zu Beginn schneller durch den *Icp* berechnet, nach vielen Iterationen kann sich eventuell aber auch der *Icp* als schneller herausstellen. Es zeigt sich, dass die Translations-Komponenten durch

den IMRP Algorithmus nur dann beeinflusst werden, wenn die Nachbarschaft zur Korrespondenzbestimmung klein genug wird. Zu Beginn wenn B_θ noch groß ist, tendiert der Algorithmus dazu, Translationen durch Rotationen auszugleichen. Dies beeinträchtigt natürlich die Stabilität des Verfahrens.

Kombination der beiden Korrespondenzen Der IMRP sei hier jedoch nur am Rande erwähnt. Eigentlich interessiert uns die Kombination der beiden Regeln *nächster-Punkt* und *gleiche-Entfernung*, um sowohl die Konvergenzgeschwindigkeit des IMRPs und die Stabilität des *Icps* zu erhalten. Dadurch erhalten wir den *Iterative Dual Correspondence* (Idc) Algorithmus:

- transformiere jeden Punkt p'_i von S_{new} in das Referenzsystem von S_{ref} durch die Näherung q_k mit $c_i = q_k(p'_i)$.
- bestimme für jeden Punkt c_i den nächstgelegenen Punkt p_j aus dem Referenz-Scans S_{ref} .

$$\min\{d_{closest-point}(c_j, p_j)\}$$

mittels der nächster-Punkt-Regel. Das Ergebnis ist eine Menge C_1 mit n Korrespondenzen (p_j, c_j) .

- bestimme für jeden Punkt c_i den nächstgelegenen Punkt p'_j aus dem Referenz-Scans S_{ref} .

$$\min\{d_{matching-range}(c_j, p_j)\}$$

mittels der gleicher-Abstand-Regel. Das Ergebnis ist eine Menge C_2 mit n Korrespondenzen (p'_j, c_j) .

- bestimme die Transformation $q_{i1} = (x_1, y_1, \theta_1)$, die den quadratischen Fehler zwischen den Korrespondenzen aus C_1 minimiert.
- bestimme die Transformation $q_{i2} = (x_2, y_2, \theta_2)$, die den quadratischen Fehler zwischen den Korrespondenzen aus C_2 minimiert.

- q_i ist somit gegeben durch $q_i = (x_1, y_1, \theta_2)$.
Wenn das Verfahren konvergiert, so ist q_{min} die gesuchte Näherung der Transformation. Wenn nicht, so iteriert man ein weiteres mal mit $q_{k+1} = q_{min}$.

Die Grundidee dieses Algorithmus ist es, die Translations-Komponenten von der nächsten-Punkt-Regel zu nehmen, wohingegen die Rotationskomponente mit der gleichen-Abstand-Regel bestimmt wird. Das Verhalten des Idcs ist laut [LM94] nun unabhängig vom Parameter B_θ .

Abbildung 2.2 zeigt die Zuordnungen der beiden Verfahren vergleichsweise bei Rotation und Translation. Die Linien stellen den Verbindungsvektor zwischen den Punkten c_i und dem Modell der Welt dar. Man sieht anhand der Vektoren deutlich, dass die Korrespondenzen in c) eine Rotation ausdrücken. In d) sieht man jedoch auch die Schwachstelle der neuen Korrespondenzzuordnung - eine Translation wird nur sehr schwer erkannt.

2.3 Metric Based Iterative Closest Points

Idee

Das Kernproblem der iterativen Scanmatching Verfahren ist die Bestimmung der Punkt-Korrespondenzen. In den beiden vorangegangenen Verfahren *Icp* und *Idc* wurde hierzu jeweils der euklidische Abstand verwendet. Darin liegt jedoch auch der Schwachpunkt der beiden Verfahren. Der *Icp* hat das Problem, dass durch eine einfache Zuordnung der räumlich nächsten Punkte Rotationen nur sehr schwer ausgedrückt werden kann, da weiter vom Sensor entfernte Punkte einen größeren Abstand zueinander aufweisen als nahe gelegene Punkte. Der *Idc* versucht dies durch zwei unterschiedliche Korrespondenz-Zuordnungen auszugleichen. Der Nachteil ist jedoch, dass für jede Iteration die Korrespondenzen und deren Transformation doppelt berechnet werden müssen.

Die Idee des Metric Based Iterative Closest Points (MbIcp) Verfahrens ist es, durch die Wahl einer anderen Metrik als des euklidischen Abstandes die Rotation gleichberechtigt zur Translation zu behandeln. Die Berechnung der Korrespondenzen und die Bestimmung der zugehörigen Translation erfolgt dann in einem einzigen Berechnungsschritt. Durch

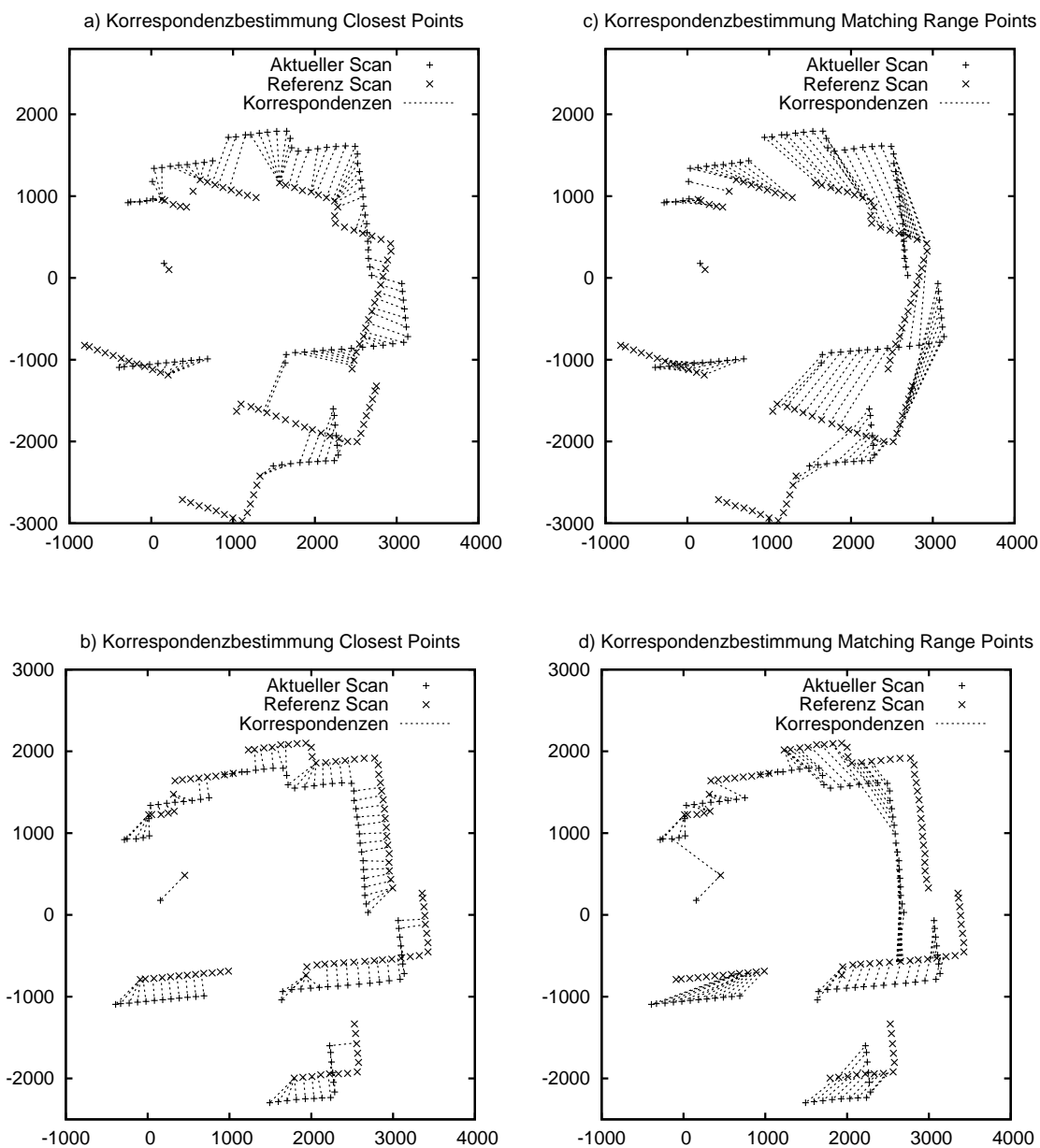


Bild 2.4: Idc: Korrespondenzzuordnungen bei Rotation und Translation. Reduktion der Scanpunkte auf 10 cm Abstände. a) Rotation um -25° Grad, Closest-Points-Regel; b) Translation um 30 cm, Closest-Points-Regel; c) Rotation um -25° Grad, Matching-Range-Points-Regel; d) Translation um 30 cm, Matching-Range-Points-Regel.

iteratives Bestimmen der Korrespondenzen konvergiert der Algorithmus wie die beiden vorhergegangenen Verfahren gegen die gesuchte Transformation.

Umsetzung der neuen Metrik

Abstand Punkt zu Punkt Eine Transformation in der Ebene ist definiert durch den Vektor $q = (x, y, \theta)$, wobei x die Translation in x-Richtung, y die Translation in y-Richtung und θ die Rotation um den Ursprung darstellt. Die Norm von q definieren wir durch

$$\|q\| = \sqrt{x^2 + y^2 + L^2\theta^2} \quad (2.8)$$

, mit $L \in \mathbb{R}$ und (homogen zu einer Länge) abhängig von der Größe der Eingabedaten. L legt fest, wie stark die Rotation im Verhältnis zu den Translationen gewichtet wird.

Betrachtet man zwei Punkte $p_1 = (p_{1_x}, p_{1_y})$ und $p_2 = (p_{2_x}, p_{2_y})$ im \mathbb{R}^2 , so definieren wir die Distanz zwischen den beiden Punkten als:

$$d_p(p_1, p_2) = \min\{\|q\| \text{ so dass } q(p_1) = p_2\} \quad (2.9)$$

mit

$$q(p_1) = \begin{pmatrix} x + \cos\theta p_{1_x} - \sin\theta p_{1_y} \\ y + \sin\theta p_{1_x} + \cos\theta p_{1_y} \end{pmatrix} \quad (2.10)$$

Die oben genannte Distanz lässt sich jedoch nicht in abgeschlossener Form darstellen. Wir können jedoch eine Annäherung berechnen, die gültig ist solange die Norm von q nicht zu groß ist. Hierzu linearisieren wir die Gleichung 2.10 mit $\theta = 0$. Die Menge der Transformationen $q(p_1) = p - 2$ kann angenähert werden durch die Lösungsmenge des folgenden Systems:

$$\begin{aligned} x + p_{1_x} - \theta p_{1_y} &= p_{2_x} \\ y + \theta p_{1_x} + p_{1_y} &= p_{2_y} \end{aligned}$$

Es gibt unendlich viele Lösungen für dieses Gleichungssystem.. Sie lassen sich ausdrücken durch:

$$\begin{aligned} x &= p_{2_x} - p_{1_x} + \theta p_{1_y} \\ y &= p_{2_y} - p_{1_y} - \theta p_{1_x} \end{aligned}$$

, wobei θ die Lösungsmenge parametrisiert.

Nach Gleichung 2.9 suchen wir ein $q = (x, y, \theta)$, so dass die Norm von q minimal ist. Setzen wir die oben berechneten Werte für x und y in Gleichung 2.8 ein, so erhalten wir für eine gegebene Rotation θ :

$$\|q\| = (\delta_x + \theta p_{1y})^2 + (\delta_y - \theta p_{1x})^2 + L^2 \theta^2$$

mit $\delta_x = p_{2x} - p_{1x}$ und $\delta_y = p_{2y} - p_{1y}$. Durch quadrieren und weiteres Umformen dieser Gleichung erhält man ein Polynom 2. Grades von θ :

$$\|q\|^2 = a\theta^2 + b\theta + c$$

mit $a = p_{1y}^2 + p_{1x}^2 + L^2$, $b = 2(\delta_x p_{1y} - \delta_y p_{1x})$ und $c = \delta_x^2 + \delta_y^2$. Da a immer größer als 0 ist, existiert nur ein Minimum für $\theta = -b/(2a)$. Somit erhalten wir:

$$\begin{aligned} \|q\|^2 &= \frac{-b^2 + 4ac}{4a} \\ &= \dots \\ &= \delta_x^2 + \delta_y^2 - \frac{(\delta_x p_{1y} - \delta_y p_{1x})^2}{p_{1y}^2 + p_{1x}^2 + L^2} \end{aligned}$$

Für den Abstand zwischen zwei Punkten erhalten wir somit folgende Approximation:

$$d_p^{app}(p_1, p_2) = \sqrt{\delta_x^2 + \delta_y^2 - \frac{(\delta_x p_{1y} - \delta_y p_{1x})^2}{p_{1y}^2 + p_{1x}^2 + L^2}} \quad (2.11)$$

Es ist zu beachten, dass diese Distanz stets kleiner als die euklidische Distanz ist. Dies kommt daher, dass der letzte Bruch in der Gleichung immer größer Null ist. Wäre er gleich Null, so würde die Gleichung der der euklidischen Distanz entsprechen. Diese kann man auch mit unserer neuen Metrik approximieren, indem man den Parameter L sehr groß wählt.

Um die Eigenschaften der neuen Distanz besser zu verstehen, kann man ihre Iso-Kurven betrachten. Die Iso-Kurve in Abbildung 2.3 verdeutlicht das Abstandsmaß unserer Approximation. Jeder Punkt auf der Ellipse hat nach unserer neuen Metrik den gleichen Abstand

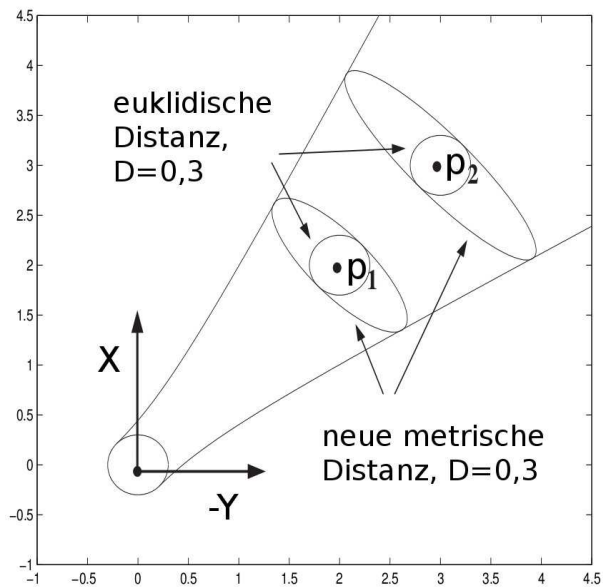


Bild 2.5: MbIcp: Iso-Kurven der neuen Metrik im Vergleich zu den Iso-Kurven des euklidischen Abstands. Jeder Punkt der Ellipse hat in seiner Metrik den gleichen Abstand zum Punkt in seinem Zentrum. Quelle: [MLM05]

zum Punkt p_1 . Diese Ellipsen sind stets in p_1 zentriert und ihre Hauptachsen entsprechen (p_{1x}, p_{1y}) und $(-p_{1y}, p_{1x})$ mit den Längen c und $c\sqrt{1 + \frac{\|p_1\|^2}{L^2}}$ mit

$$\{p_2 \in R^2 \mid d_p^{app}(p_1, p_2) = c\}$$

Desto weiter der Punkt p_1 vom Ursprung des Koordinatensystems entfernt ist, desto mehr streckt sich die Ellipse seiner Iso-Kurve. Dadurch wird rechnerisch die Distanz verringert, die ein Punkt nach einer Rotation von seinem Korrespondenzpunkt entfernt ist. Für eine Translation hingegen nähert sich die metrische Distanz der euklidischen an. Die Kreise um die Punkte p_1 zeigen vergleichsweise die Iso-Kurve der euklidischen Distanz an.

Abstand Punkt zu Linie Wiederum möchten wir den Abstand von einem Punkt p_1 und einem Liniensegment l betrachten. Das Liniensegment l ist definiert durch die beiden Start- und Endpunkte s_1 und s_2 . Der Abstand zwischen p_1 und dem Segment $[s_1, s_2]$ ist gegeben durch den Abstand zwischen p_1 und dem Punkt des Liniensegmentes l , welcher dem Punkt am nächsten liegt:

$$d_{pl}(p_1, [s_1, s_2]) = \min_{\lambda \in [1,0]} d_p(p_1, (1 - \lambda)s_1 + \lambda s_2) \quad (2.12)$$

Verwenden wir wieder unsere eigens definierte Metrik 2.8 und wenden die gleichen Annahmen wie zuvor an, so gelangen wir zu folgender Näherung des Abstandes:

$$d(\lambda)^2 \approx \delta_x(\lambda)^2 + \delta_y(\lambda)^2 - \frac{(\delta_x(\lambda)p_{1y} - \delta_y(\lambda)p_{1x})^2}{p_{1y}^2 + p_{1x}^2 + L^2} \quad (2.13)$$

wobei

$$d(\lambda) = (\delta_x(\lambda), \delta_y(\lambda) = s_1 - p_1 + \lambda(s_2 - s_1)) \quad (2.14)$$

Setzen wir Gleichung 2.14 in Gleichung 2.13 ein, so erhalten wir wieder ein Polynom 2. Grades von λ :

$$d(\lambda)^2 \approx a\lambda^2 + b\lambda + c$$

mit den folgenden Koeffizienten:

$$\begin{aligned} a &= u_{2_x}^2 + u_{2_y}^2 - \frac{p_{1_x} u_{2_x} - p_{1_x} u_{2_y}}{p_{1_x}^2 + p_{1_y}^2 + L^2} \\ b &= 2(u_{2_x} \delta_{1_x} + u_{2_y} \delta_{1_y}) - 2 \frac{(p_{1_y} u_{2_x} - p_{1_x} u_{2_y})(\delta_{1_x} p_{1_y} - \delta_{1_y} p_{1_x})}{p_{1_x}^2 + p_{1_y}^2 + L^2} \\ c &= \delta_x^2 + \delta_y^2 - \frac{(\delta_x p_{1_y} - \delta_y p_{1_x})^2}{p_{1_y}^2 + p_{1_x}^2 + L^2} \end{aligned}$$

mit $u_2 = (u_{2_x}, u_{2_y}) = s_2 - s_1$ und $d_1 = (d_{1_x}, d_{1_y}) = s_1 - p_1$. Da der Koeffizient a immer positiv ist existiert nur ein einziges Minimum für $\lambda = \frac{-b}{2a}$. Somit ergibt sich eine Approximation für den Abstand $d_{pl}(p_1, [s_1, s_2])$ für nicht zu große Rotationen:

$$d_{pl}(p_1, [s_1, s_2]) \approx \begin{cases} d_p(p_1, s_1) & \text{für } \lambda < 0 \\ d_p(p_1, s_2) & \text{für } \lambda > 1 \\ \sqrt{\frac{-b^2 + 4ac}{4a}} & \text{für } 0 \leq \lambda \leq 1 \end{cases}$$

Der nächste Punkt zu p_1 ist demnach s_1, s_2 , oder $s_1 - \frac{b}{2a} u_2$.

Minimierung der kleinsten quadratischen Abstands Um das q zu bestimmen, welches die Kriterien von 2.2 erfüllen, führen wir zunächst folgende Notation ein: Die Koordinaten der Referenz-Punkte p_i werden repräsentiert durch (p_{ix}, p_{iy}) und die der transformierten aktuellen Punkte $q_k(p'_i) = c_i$ durch (c_{ix}, c_{iy}) . Durch Verwendung der genäherten Distanz 2.11 erhalten wir folgende Gleichung:

$$E_{dist}(q) = \sum_{i=1}^n \left(\delta_x^2 + \delta_y^2 - \frac{(\delta_x p_{1_y} - \delta_y p_{1_x})^2}{p_{1_x}^2 + p_{1_y}^2 + L^2} \right) \quad (2.15)$$

mit

$$\begin{aligned} \delta_{ix} &= c_{ix} - c_{iy} \theta + x - p_{ix} \\ \delta_{iy} &= c_{ix} \theta + c_{iy} + y - p_{iy} \end{aligned}$$

Gleichung 2.15 ist bezüglich q quadratisch.

$$E_{dist}(q) = q^T A_q + 2b^T q + c$$

wobei c eine konstante Zahl ist. A ist eine symmetrische Matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix}$$

mit

$$\begin{aligned} a_{11} &= \sum_{i=1}^n 1 - \frac{p_{iy}^2}{k_i} \\ a_{12} &= \sum_{i=1}^n \frac{p_{ix}p_{iy}}{k_i} \\ a_{13} &= \sum_{i=1}^n -c_{iy} + \frac{p_{iy}}{k_i}(c_{ix}p_{ix} + c_{iy}p_{iy}) \\ a_{22} &= \sum_{i=1}^n 1 - \frac{p_{ix}^2}{k_i} \\ a_{23} &= \sum_{i=1}^n c_{ix} + \frac{p_{ix}}{k_i}(c_{ix}p_{ix} + c_{iy}p_{iy}) \\ a_{33} &= \sum_{i=1}^n c_{ix}^2 + c_{iy}^2 + \frac{1}{k_i}(c_{ix}p_{ix} + c_{iy}p_{iy}^2) \end{aligned}$$

und

$$B = \begin{pmatrix} \sum_{i=1}^n c_{ix} - p_{ix} - \frac{p_{iy}}{k_i}(c_{ix}p_{ix} + c_{iy}p_{iy}) \\ \sum_{i=1}^n c_{iy} - p_{iy} - \frac{p_{ix}}{k_i}(c_{ix}p_{ix} + c_{iy}p_{iy}) \\ \sum_{i=1}^n \left[\frac{1}{k_i}(c_{ix}p_{ix} + c_{ix}p_{iy}) - 1 \right] (c_{ix}p_{iy} + c_{iy}p_{ix}) \end{pmatrix}$$

mit $k_i = p_{1x}^2 + p_{1y}^2 + L^2$. Das q welches $E_{dist}(q)$ minimiert ist dann gegeben durch

$$q_{min} = -A^{-1}b.$$

Anmerkung d_p nach 2.8 ist eine echte metrische Distanz, da sie die folgenden Eigenschaften erfüllt:

- $d_p(p_1, p_2) = d_p(p_2, p_1)$

- $d_p(p_1, p_2) = 0 \Rightarrow p_1 = p_2$
- $d_p(p_1, p_2) + d_p(p_2, p_3) \geq d_p(p_1, p_3)$

Für d_p^{app} nach 2.11 trifft dies jedoch nicht zu, was leicht zu beweisen ist. Das Problem der Approximation ist, dass die Koordinaten des ersten Punktes p_1 die metrischen Eigenschaften beeinflussen. Durch andere Entfernung oder die Lage in einem anderen Winkelbereich vom Ursprung aus gesehen verändern sich die zugehörigen Iso-Kurven. Somit sind die erste und die dritte der oben genannten Eigenschaften nicht erfüllt.

Da man jedoch davon ausgehen kann, dass im Kontext des Scanmatchings die Differenz zwischen p_1 und p_2 relativ gering ist, ist der auftretende Fehler akzeptabel.

Umsetzung des Verfahrens

Dieses Verfahren wird bis zur Konvergenz wiederholt:

- Zunächst transformiert man jeden Punkt p'_i von S_{new} in das Referenzsystem von S_{ref} durch die Näherung q_k mit $c_i = q_k(p'_i)$. Nun bestimmt man für jeden Punkt c_i den nächstgelegenen Punkt p_j aus dem Referenz-Scans S_{ref} .

$$\min\{d_p^{app}(c_j, p_j)\} \quad (2.16)$$

Das Ergebnis ist eine Menge C mit n Korrespondenzen (p_j, c_j) .

- Dann bestimmt man die Näherung für die Transformation, die den quadratischen Fehler zwischen den Korrespondenzen aus C minimiert. Hierzu minimiert man die folgende Gleichung über q :

$$E_{dist}(q) = \sum_{i=1}^n d_p^{app}(r_j, q(c_j))^2 \quad (2.17)$$

Wenn das Verfahren konvergiert, so ist q_{min} die gesuchte Näherung der Transformation. Wenn nicht, so iteriert man ein weiteres mal mit $q_{k+1} = q_{min}$.

Die Effekt der metrischen Distanz ist in Abbildung 2.3 zu sehen. Dort sind Korrespondenzzuordnungen anhand eines exemplarischen Laserscans dargestellt. In Abbildung 2.3 a) ist deutlich zu erkennen, dass die weit vom Ursprung entfernten Korrespondenzen die Bewegung einer Rotation andeuten. Die Punkt-Korrespondenzen nach der Translation 2.3 b) ähneln jedoch sehr denen, die auch mit euklidischem Abstand ermittelt werden.

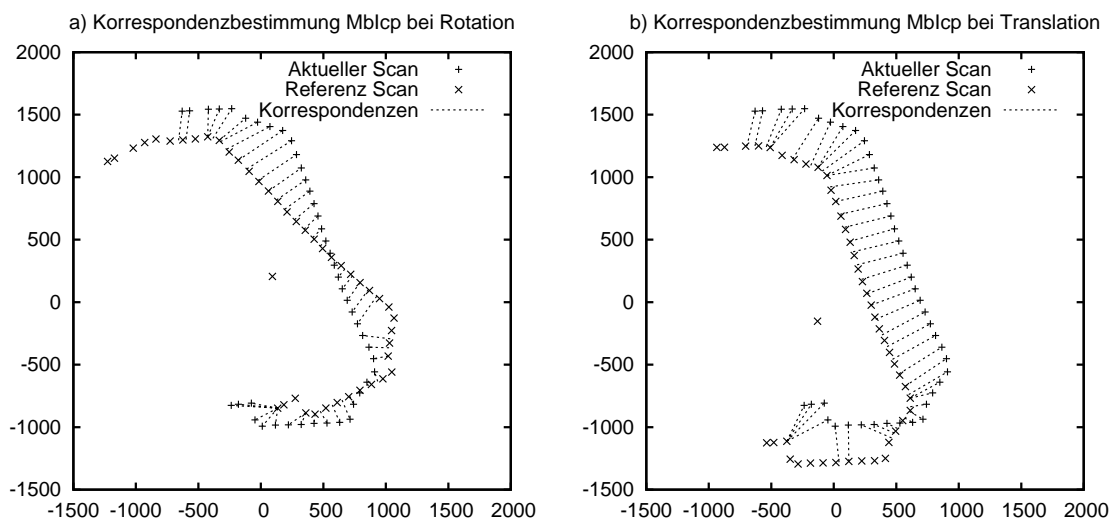


Bild 2.6: Mblcp: Korrespondenzzuordnungen bei Rotation und Translation. Reduktion der Scanpunkte auf 10 cm Abstände. a) Rotation um 25° Grad; b) Translation um -30 cm in x- und y-Richtung.

Kapitel 3

Eigener Ansatz

3.1 Implementation

Die Implementation der Scanmatching Verfahren erfolgt weitgehend analog zur in Kapitel: *sota* vorgestellten Theorie. Zum effizienten Arbeiten der Algorithmen sind jedoch ein paar Ergänzungen notwendig, welche in diesem Kapitel vorgestellt werden.

3.1.1 Konvergenzkriterien

Bei den drei Scanmatching Verfahren handelt es sich um iterative Algorithmen. Sind die Kriterien zur Korrespondenzbestimmung und Minimierung des quadratischen Fehlers gut gewählt, so konvergieren die Verfahren. Die Konvergenz wird als erreicht betrachtet, sobald die aktuell berechnete Teiltransformation einen gewissen Schwellwert unterschreitet. Hierbei werden die x -, y - und θ -Komponenten getrennt betrachtet. Die Wahl der Schwellwerte beeinflusst die Anzahl der benötigten Iterationen und die Genauigkeit des Ergebnisses.

3.1.2 Fehlerfall

Konvergiert das Scanmatching-Verfahren, so hat man im Normalfall die optimale Transformation zwischen Referenz- und aktuellem Scan gefunden. Es ist jedoch möglich, dass das Verfahren in einem lokalen Minima konvergiert.

Für diesen Fall berechnen wir einen Fehlerwert E_{dist} , der aus der verbleibenden Diskrepanz der Scans nach Anwendung der Transformation bestimmt wird. Hierzu betrachten wir lediglich die zuvor verwendeten Punktkorrespondenzen um eventuelle Ausreißer zu ignorieren. Denn durch den unterschiedlichen Aufnahmeort und -zeitpunkt ist es nicht garantiert, dass sich die Scans vollständig überdecken und somit jeder Scanpunkt einen Korrespondenzpartner besitzt. Den Fehlerwert definieren wir nach Gleichung 2.2 als den mittleren euklidischen Abstand zwischen den Korrespondenzen.

Bei Anwendung des Idcs haben wir stets zwei mögliche Mengen an Korrespondenzpunkten. Zur Identifizierung der fehlerhaften Transformationen verwenden wir die Korrespondenzen der *nächsten-Punkt-Regel*. Diese entsprechen intuitiv eher der Güte der Transformation als die Korrespondenzen der *gleichen-Abstand-Regel*. Außerdem wird durch die kontinuierliche Verringerung des zugelassenen Winkelabstandes θ_c der Korrespondenzpartner in der *gleichen-Abstands-Regel* der mögliche Fehlerwert E_{dist} indirekt verringert, was das Ergebnis verfälscht.

Außerdem möchten wir feststellen, ob sich die Scans nach Anwendung der Transformation ausreichend überdecken. Dies überprüfen wir mittels der Anzahl der zuletzt gefundenen Korrespondenzen und vergleichen diese mit einem Schwellwert.

Um die gefundene Transformation als gültig anzuerkennen, erwarten wir einen Fehlerwert von kleiner als 10 cm unter der Verwendung von mindestens $2/3$ der Scanpunkte. Dies sind sehr großzügig gewählte Parameter, jedoch haben sie sich in der Praxis als ausreichend herausgestellt. Konvergiert der Algorithmus in einem lokalen Minimum, so sind meisten bereits die Ausgangsdaten ungünstig und die bestimmte Transformation grundlegend falsch. Auch wird der Algorithmus nach einem maximalen Durchlauf von 80 Schritten abgebrochen, da eine solch lange Laufzeit ebenso auf grundlegend falsche Korrespondenzbildungen hinweist.

3.1.3 Klassenstruktur

Der Aufbau des ScanMatcher-Packages ist wie folgt: Es gibt eine abstrakte Oberklasse ScanMatcher, welche die Implementation der notwendigen Funktionen für ein iteratives Scanmatching erzwingt. Öffentlich sichtbar sind hiervon lediglich die Methoden zur Berechnung der Transformation zwischen zwei Laserdaten. Diese ermöglichen das Abbilden sowohl zweier Punktwolken aufeinander als auch einer Punktwolke auf eine Linienstruktur im zwei-dimensionalen Raum. Desweiteren muss jeder ScanMatcher eine Funktion bereitstellen, welche eine optimale Transformation zwischen Punktkorrespondenzen berechnen kann. Die einzige Funktion, die für alle ScanMatcher gleich implementiert ist, ist der Funktion zur Berechnung des Fehlerwertes bei gegebenen Punktkorrespondenzen. Alle Parameter, die zum Scanmatching benötigt werden, sind Klassenattribute und nur über Setter-Methoden zu ändern.

Der Icp-ScanMatcher und der MbIcp-ScanMatcher sind von der abstrakten Oberklasse ScanMatcher abgeleitet. Sie stellen jeweils eine Funktion zur Berechnung der Punktkorrespondenzen. Der MbIcp-ScanMatcher benötigt außerdem Methoden zur Berechnung der ihm eigenen metrischen Distanz. Der Idc-ScanMatcher ist wiederum vom Icp-ScanMatcher abgeleitet, da er eine Spezialisierung dessen darstellt. Er besitzt eine weitere Methode zur Berechnung von Punktkorrespondenzen, welche mit der *gleichen-Abstands-Regel* ermittelt werden. Der Aufbau des ScanMatcher-Paketes ist anhand von Abbildung 3.1 nachzuvollziehen.

Die Klasse der Scanmatcher Utils enthält wichtige Hilfsfunktionen zur Verarbeitung von Laserdaten. Diese werden sowohl von den Scanmatchern selbst als auch von den aufrufenden Klassen genutzt.

Die Implementation kann als Strategy-Pattern gelesen werden. Der ScanMatcher ist die abstrakte Strategie; der Icp-, Idc- und MbIcp-ScanMatcher stellen jeweils Implementationen dieser dar. Die Wahl der konkreten Strategie ist dem aufrufenden Programm überlassen. Zur Durchführung des Scanmatchings gibt dieses die zu matchenden Daten mit dem ausgewählten ScanMatcher an die Utils-Klasse. Diese stellt eine Funktion zum Scanmatching mit integrierter Datenvorverarbeitung zur Verfügung. Auf diese Art und Weise ist die Wahl des Matching-Algorithmuses von seiner Ausführung und Ablauflogik entkoppelt.

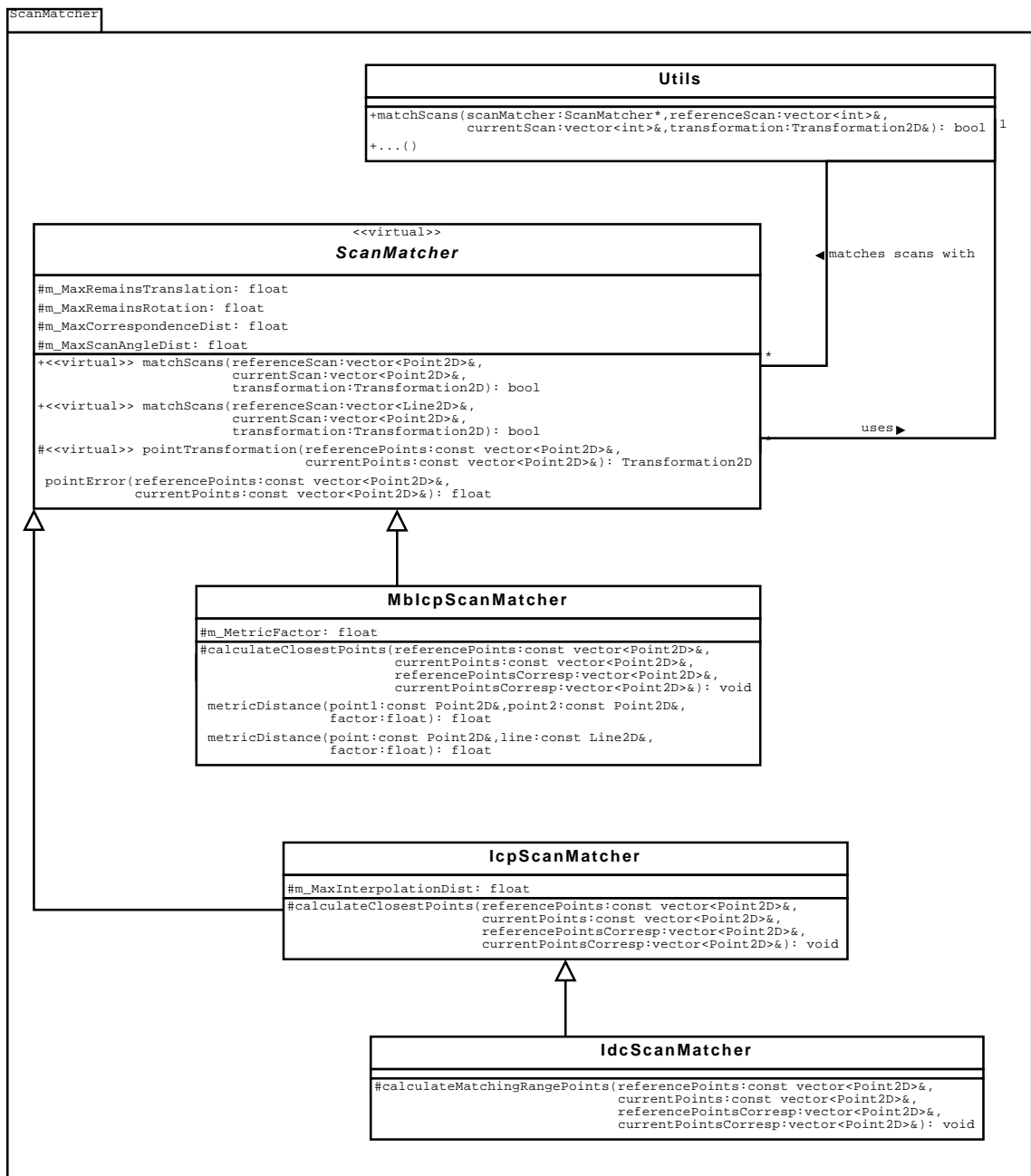


Bild 3.1: Klassendiagramm: Scan-Matcher Paket

3.1.4 Ablauf Scanmatching

Nach der Integration der Konvergenzkriterien und des Fehlerfalls ergibt sich somit folgender Ablauf für iterative Scanmatching, wie er in Abbildung 3.2 dargestellt wird. Dies entspricht dem Vorgehen, welches für *Icp* und *MbIcp* verwendet wird. Das Vorgehen des *Idc* müsste noch jeweils um eine weitere Korrespondenz- und Transformationsbestimmung erweitert werden. Die Transformation ergibt sich dann jeweils aus den beiden Teiltransformationen für Rotation und Translation.

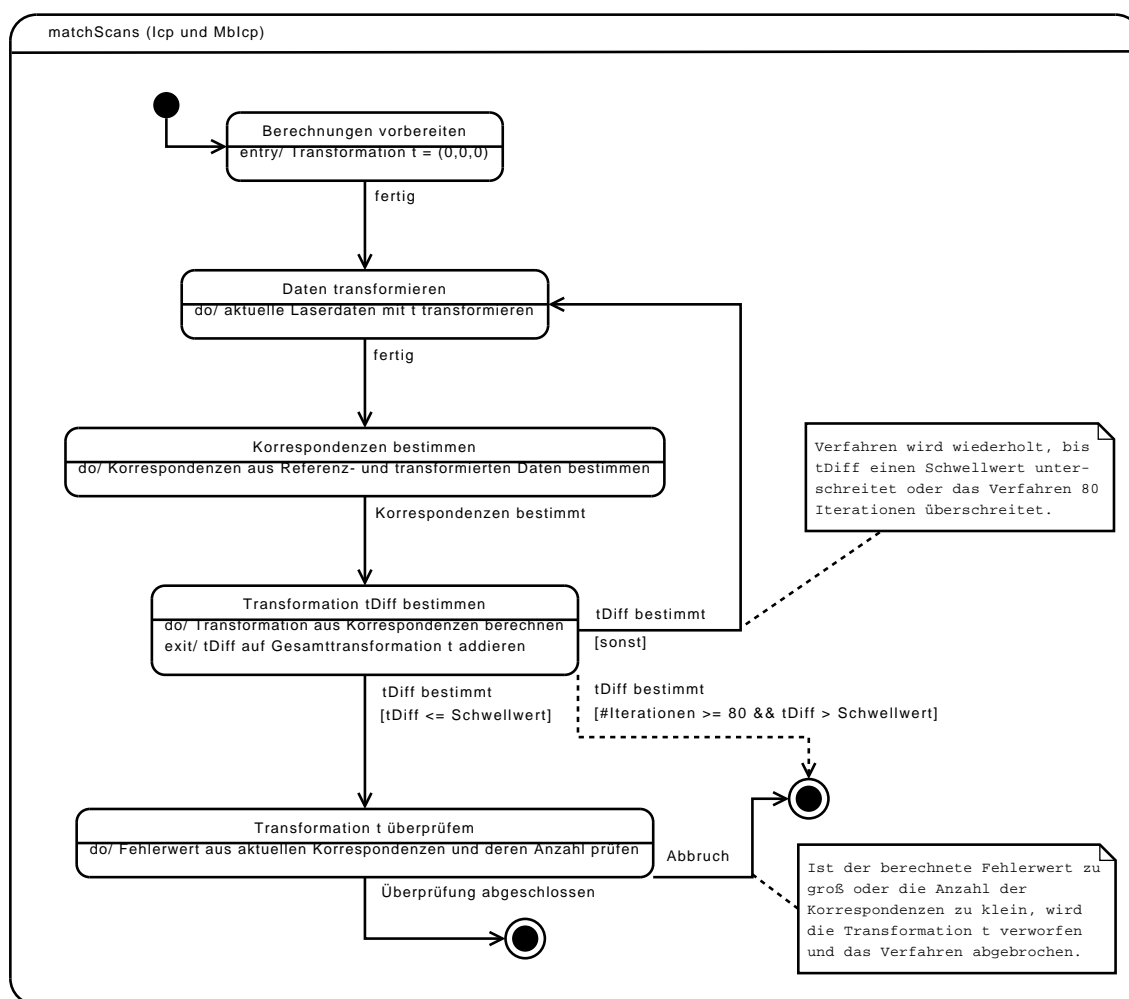


Bild 3.2: StateChart: Ablauf des Scanmatchings für *Icp* und *MbIcp*.

3.1.5 Entfernen falscher Punkt Korrespondenzen

Natürlich ist es möglich, dass bei der Korrespondenzbestimmung auch falsche Korrespondenzen ermittelt werden. Diese beeinflussen das Gesamtergebnis negativ in Qualität und Performanz. Daher könnte man es als sinnvoll betrachten, falsche Korrespondenzen zu identifizieren und zu nicht zur Transformationsbestimmung zu verwenden.

Ein einfaches Verfahren bietet das Ignorieren der am weitesten voneinander entfernten Korrespondenzpartner. Hierzu sortiert man die gefundenen Korrespondenzen nach ihrer Entfernung zueinander und verwendet dann lediglich die ersten zur Korrespondenzbestimmung. Zum Beispiel könnte man so die letzten 20% ignorieren. Dieses Verfahren hat durch das Sortierungsverfahren einen zusätzlichen Aufwand von $\mathcal{O}(n \cdot \log n)$ und bewährt sich nicht in allen Scanmatching Algorithmen. Näheres hierzu siehe in Kapitel 4.

3.2 Vorverarbeitung der Laserdaten

Für ein effizientes und intelligentes Verhalten der Matching-Algorithmen ist es von Nöten, die Laserdaten vorzubereiten. Im folgenden werden einige Filter-Verfahren vorgestellt, die helfen sollen die Qualität des Scanmatching Verfahren zu heben. Die Abbildung 3.2 zeigt den sequentiellen Ablauf der Verarbeitungsschritte. Hierbei wird zwischen der Verarbeitung von Punkten und Linien unterschieden.

3.2.1 Entfernen der Ausreißer

Zu Beginn der Verarbeitung der Laserdaten bietet es sich an diese nach Ausreißern zu filtern. Das Problem ist, dass einzelne Ausreißer durch die Punkt-Reduzierung ein zu großes Gewicht erhalten würden. Der verwendete Filter muss lediglich die Eigenschaft haben, dass er die Ecken und Kanten möglichst gut erhält. Da es sich anbietet die Laserdaten ebenfalls für den Einsatz in der Kartenerstellung zu filtern, wird die Filterung bereits zu einem früheren Zeitpunkt im PartikelFilter durchgeführt. Bei dem dort verwendeten Filter handelt es sich um einen einfacher Ausreißer-Filter, der die Sprünge in den Laser-Daten überwacht und Ausreißerpunkte als Fehlerwerte markiert.

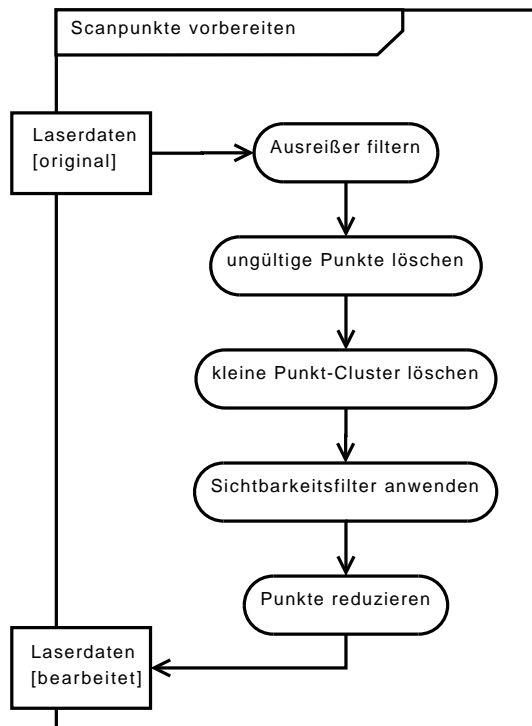


Bild 3.3: Aktivitätsdiagramm: Vorbereitung der Scan-Punkte

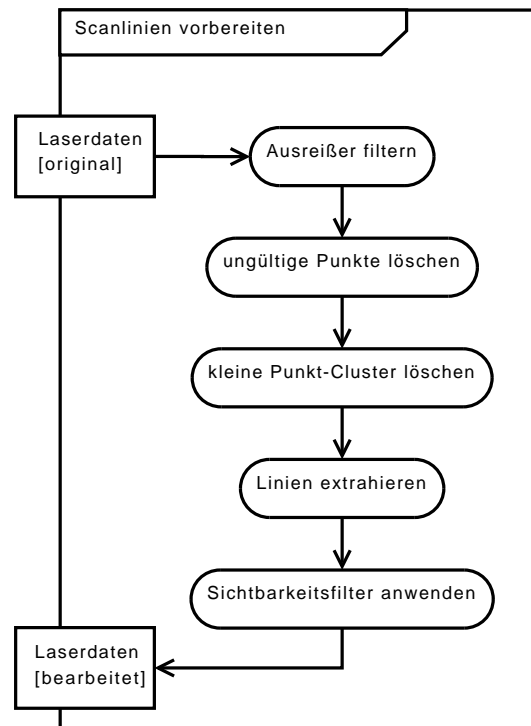


Bild 3.4: Aktivitätsdiagramm: Vorbereitung der Scan-Linien

3.2.2 Löschen ungültiger Punkte

Es ist möglich, dass der Laserscanner ungültige Werte liefert. Das kann zum Beispiel dadurch verursacht werden, dass sich Schmutz auf dem Laser-Scanner befindet oder die Messungen in bestimmten Winkelbereichen generell Fehlerwerte liefern. Desweiteren ist es möglich, dass der Scanner aus Versehen Teile des Roboters mitscannt, wie zum Beispiel ein nicht korrekt befestigtes Kabel in der Nähe des Laserscanners.

Solch fehlerhafte Scan-Punkte, die sich mit der Bewegung des Roboters fortbewegen, können für die Scanmatching-Verfahren sehr verheerend sein. Das eigentliche Problem liegt in der Umsetzung des Reduktionsfilters. Aus Gründen der Effizienz-Steigerung geht man dort davon aus, dass die übergebenen Punkte korrekt in aufsteigender Winkel-Reihenfolge vorliegen. Implizit bedeutet dies, dass man auch davon ausgeht, dass räumlich benachbarte Punkte auch in den Laserdaten hintereinander liegen. Viele der fehlerhaften Laser-Punkte liegen zwar räumlich benachbart in der Nähe des Laser-Scanners, jedoch nicht zwingendermaßen auf aufeinanderfolgenden Positionen in den Scandaten. Daher ist es möglich, dass sie von dem Reduktionsfilter nicht erfasst werden. Je nach Parameterwahl für den Aufruf des Reduktionsfilters ist es in der engen Umgebung der RoboCup Arena realistisch, dass lediglich 40 Punkte im Scan verbleiben. Handelt es sich dabei bei 5 von ihnen um Fehlmessungen, so ist ihr Einfluss nicht zu vernachlässigen. Gerade der *Idc* zeigte sich für die fehlerhaften Scanpunkte sehr anfällig. Ohne Unterstützung durch die Odometrie-Daten tendierte er häufig zu einer viel zu geringen Translation.

Für die Funktionsweise des Scanmatchings ist es ausreichen, wenn man die Punkte löscht, die sich zu nah am Laserscanner befinden. Als geeigneten Abstand haben sich hier 20 cm herausgestellt. In diesem Radius um den Laserscanner ist es aufgrund der Bauweise des Roboters nicht möglich, dass wahre Abstandsmessungen von der Umgebung zurück gegeben werden. Abbildung 3.5 zeigt ein Anwendungsbeispiel des Filters.

3.2.3 Entfernen kleiner Punkt-Cluster

Eine sehr große Schwachstelle weist das Scanmatching auf in dynamischen Umgebungen. Denn die Scanmatching Verfahren beziehen auch diejenigen Scan-Punkte mit ein, die zu

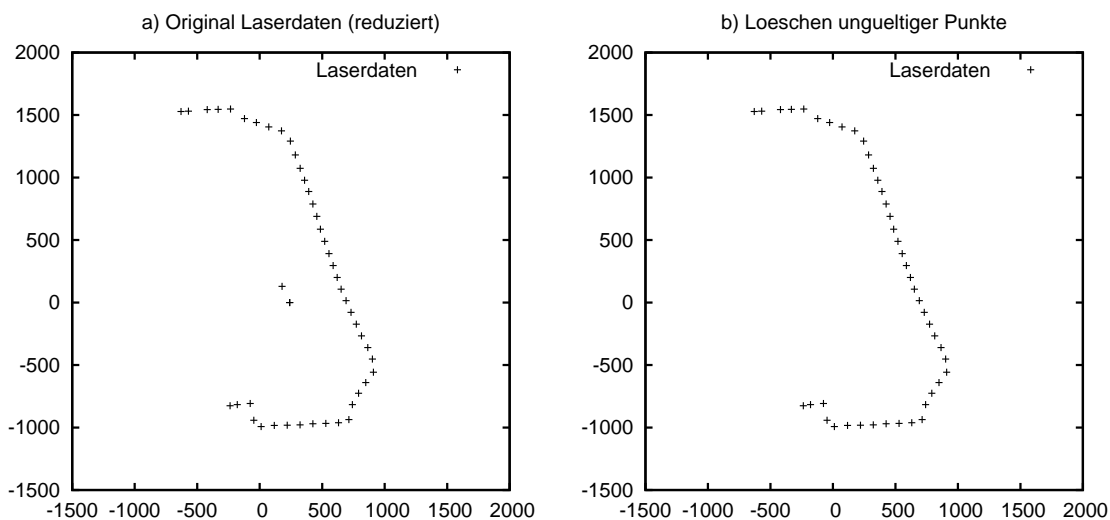


Bild 3.5: Filter: Löschen ungültiger Punkte

dem beweglichen Objekten gehören. In der @ *Home*-Arena des *RoboCups* ist dies zum Beispiel ein sehr großes Problem. Da es sich um eine möglichst realistische Haushaltsumgebung handeln soll, es es nur normal, dass sich während der Spiele Personen und andere Roboter in der Arena aufhalten.

Ein sehr einfacher Ansatz zur Behebung dieses Problems ist die Identifizierung und das Löschen kleiner Punkte-Cluster, wie zum Beispiel die der Beine. Dieser Filter ist noch verbesserungsfähig. Man muss acht geben, dass man so nicht zu viele Informationen aus den Daten entfernt.

Folgende Parameter steuern das Löschen der kleinen Punkte-Cluster.

- *maxClusterDist* der maximale Abstand zweier aufeinanderfolgender Punkte eines Clusters
- *minClusterSize* der minimal erlaubte Durchmesser eines Clusters.

Man muss acht geben, dass man so nicht zu viele Informationen aus den Daten entfernt.

3.2.4 Extraktion von Linien

Die Extraktion von Linien stellt sich als eine gute Möglichkeit dar, die Effizienz der Algorithmen zu steigern. Wenn man die Anzahl der Scan-Punkte sowieso stark reduziert um dann nur noch mit der Interpolationen dieser zu arbeiten, so kann man genau so gut Linien extrahieren und Korrespondenzpunkte auf diesen bestimmen. Sind die Parameter gut justiert und weist die Umgebung ausreichend Linien auf, so ist die Verwendung von Linien genauer als die von reduzierten Punkten. Denn die durch Interpolation der reduzierten Punkte gewonnenen Liniensegmente umfassen nicht alle Scan-Punkte, die durch eine Lineinextraktion erhalten bleiben könnten.

Abbildung 3.6 a) zeigt einen Datensatz an Scanpunkten und Abbildung 3.6 b) die daraus extrahierte Linienstruktur. Wie man sieht, wird die Struktur der Umgebung ausreichend repräsentiert. Befindet man sich jedoch in einer nicht so gut strukturierten Umgebung wie dem RoboCup, kann dadurch wertvolle Information verloren gehen.

Der hierzu verwendete Algorithmus von Gutmann und Schlegel [Gut00] arbeitet wie folgt:

- zunächst durchlaufe die alle Scanpunkte und unterteile sie in einzelne Punktwolken, wenn der Abstand zwischen zwei aufeinander folgenden Punkten zu groß sein sollte.
- für jede so gewonnene Punktwolke
 - lege eine Linie l vom Start- zum Endpunkt
 - überprüfe ob l genügend Punkte umfasst
Wenn ja, fahre fort.
Wenn nein, verwirfe l .
 - überprüfe ob die Standardabweichung aller Punkte l gering genug ist.
Wenn ja, speichere Linie l
Wenn nein, suche Punkt mit größtem Abstand zur Linie, zerteile dort die Punktwolke und starte von neuem.

Als Parameter zur Steuerung der Linienextraktion bleiben folgende Parameter:

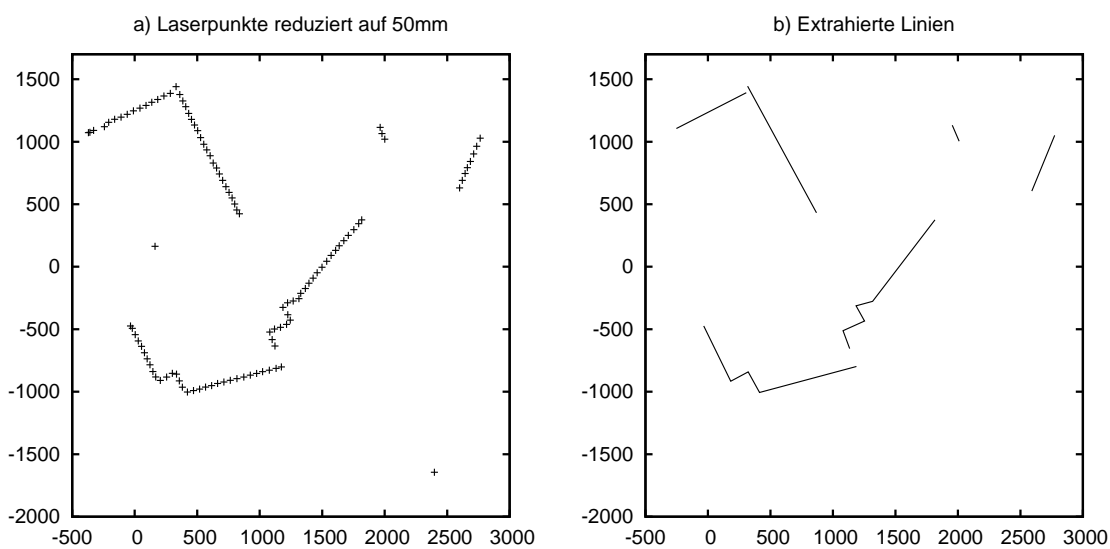


Bild 3.6: Filter: Extraktion von Linien. a) Original Laserdaten reduziert auf 50 mm Punktdichte, Anzahl 109; b) Extrahierte Linien mit maximalem Punktabstand 50 mm, minimaler Anzahl Punkte auf einer Linie 10, minimaler Linienlänge 100 mm, maximaler Standardabweichung der Punkte von zugehöriger Linie 15 mm, Anzahl 13.

- *maxPointDist* der maximale Abstand zwischen zwei Punkten auf der Linie
- *minPointsOnLine* die minimale Anzahl Punkte auf einer Linie
- *maxSigma* die maximale Standardabweichung der Punkte einer Linie von derselben.

Mit diesen Parametern kann die Anzahl und die Qualität der Linien gesteuert werden.

3.2.5 Projektions- und Sichtbarkeitsfilter

Mittels eines Projektionsfilters untersucht man für jeden Scan-Punkt eines Datensatzes, ob er von der Aufnahmeposition des anderen Scans sichtbar ist. Die zweite Aufnahmeposition muss demnach gegeben sein. Mit einem solchen Filter möchte man falsche Korrespondenzbildung unterbinden und somit die Transformationsbestimmung verbessern, beziehungsweise beschleunigen.

Lu und Millois [LM97] haben 1997 einen Algorithmus veröffentlicht, der diese Aufgabe erfüllt. Sie überprüfen für jeden gegebenen Scan-Punkt, ob er durch einen der anderen Scan-Punkte oder einem möglicherweise interpolierten Scan-Punkt verdeckt wird. Desweiteren vergleichen sie den Aufnahmewinkel, unter dem die Scans aufgenommen wurden. Somit werden zum Beispiel Wände aus den Scans entfernt, die von unterschiedlichen Seiten gescannt wurden. Dies soll verhindern, dass durch das Scanmatching Punkte aufeinander gematcht werden, die einander nicht entsprechen. Werden zum Beispiel die Punkte der unterschiedlichen Seiten einer Wand aufeinander gematcht, so hat das den Effekt, dass die Dicke der Wand in der Karte auf Null reduziert wird. Somit wird die Karte gestaucht.

Abbildung 3.7 zeigt die Wirkung des Projektionsfilters an einem Beispiel. Die obere Abbildung zeigt den original Laserscan, welcher vom Ort O_1 aufgenommen wurde. In der unteren Abbildung sind diejenigen Laserpunkte gelöscht, welche vom nächsten Aufnahmeort O_2 aus nicht zu sehen sind. Da die beiden Aufnahmeorte sehr weit voneinander entfernt sind, werden große Teile des Scans entfernt.

Leider besitzt dieses Verfahren einen quadratischen Aufwand, da jeder Punkt mit jedem verglichen werden muss. Desweiteren arbeitet er natürlich am besten, wenn er vor der Re-

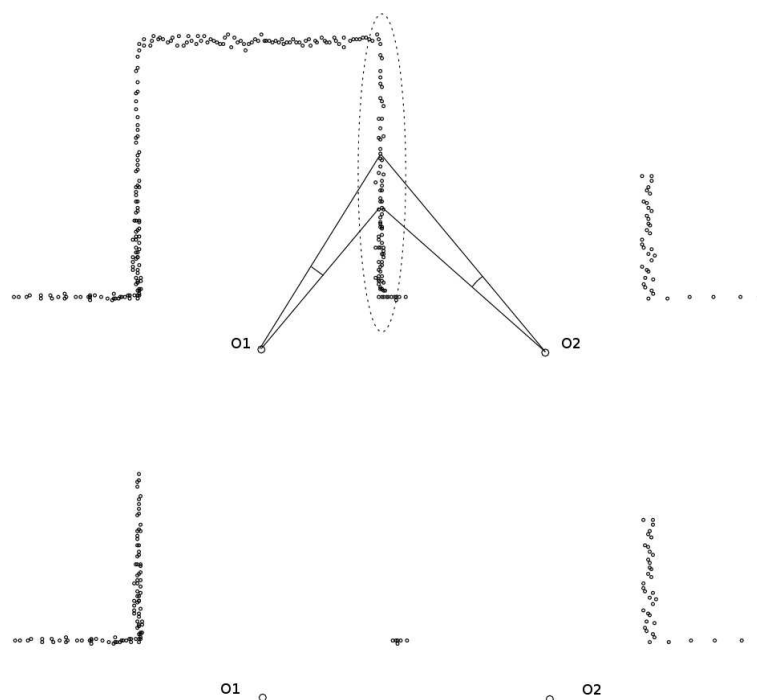


Bild 3.7: Filter: Projektionsfilter. Quelle: [Gut00].

duzierung der Punkte angewendet wird. Dies führt zu einer verhältnismäßig langen Laufzeit.

Das Anwendungsgebiet für Robbie liegt hauptsächlich in Szenarien des RoboCups. Dort können wir in der Tat von einer Wanddicke Null ausgehen, da die Dicke der Wände in der Rasterisierung unserer Karten von 5 cm verloren gehen. Da alle 0,5 Sekunden einen Mapping-Schritt durchgeführt wird, ist der Unterschied zwischen den beiden Scans und die entstehenden Differenzen nicht so groß. Dennoch tritt das Problem der plötzlich auftauchenden Wände gelegentlich auf.

Eigene Tests haben ergeben, dass die Verwendung des Projektionsfilters die Qualität der Karten kaum beeinflusst. Aus Effizienzgründen ist er daher nicht in das allgemeine Verfahren eingebunden.

Alternativ wurde jedoch ein einfacherer Sichtbarkeitsfilter entwickelt, welcher die Laserdaten auf den messbaren Bereich zuschneidet und nur linearen Aufwand benötigt.

Einfacher Sichtbarkeitsfilter auf Punkten Als Eingabe für den einfachen Sichtbarkeitsfilter auf Punkten betrachten wir die Punkte eines Laser-Scans $S = \{p_i = (x_i, y_i) | i \in N\}$ im lokalen Roboter-Koordinatensystem und eine Pose $O = \{(x, y, \theta)\}$ ausgehend vom gleichen Koordinatensystem.

Zu Beginn transformieren wir alle Scan-Punkte in das Koordinatensystem der gegebenen Pose. Dies entspricht der negativen Transformation, die die Pose repräsentiert. Auch beachten wir hier die Position des Laserscanners und berechnen die Koordinaten im Laserscanner-Koordinatensystem. Dann löschen wir all diejenigen Scan-Punkte p_i des Datensatzes, die:

- eine zu große Distanz zum Laserscanner aufweisen, also deren Betrag des Punktvektors $|\vec{p}_i|$ größer ist als die maximale Laserscanner-Reichweite L_{max}

$$|\vec{p}_i| > L_{max}$$

- im toten Winkel des Laserscanners liegen. Dies lässt sich aus den Skalarprodukten der Punktvektoren \vec{p}_i mit den Normalen derjenigen Vektoren R_l und R_r berechnen, die den toten Winkel eingrenzen

$$\vec{p}_i \cdot R_l > 0 \text{ und } \vec{p}_i \cdot R_r < 0$$

Abbildung 3.8 zeigt, wie der Filter einen Laser-Scan auf die neue Pose zurechtschneidet.

Einfacher Sichtbarkeitsfilter auf Linien Der einfache Sichtbarkeitsfilter auf Linien basiert auf dem gleichen Prinzip wie der auf Punkten. Als Eingabe haben wir Linien die aus einem Laser-Scans extrahiert wurden $L = \{l_i = (p_j, p_{j+1}) | i, j \in N, s \in S\}$ im lokalen Roboter-Koordinatensystem und eine Pose $O = \{(x, y, \theta)\}$ ausgehend vom gleichen Koordinatensystem.

Zu Beginn transformieren wir wieder alle Scan-Linien in das Laserscanner-Koordinatensystem unter Berücksichtigung der gegebenen Pose. Nun schneiden wir all diejenigen Scan-Linien l_i des Datensatzes, welche den toten Winkel des Laserscanners schneiden oder seine maximale Reichweite verlassen. Auch in diesem Fall geschieht dies mittels Skalarprodukt in linearer Laufzeit. In Abbildung 3.9 sieht man, wie der Laser-Scan auf den sichtbaren Bereich beschnitten wird.

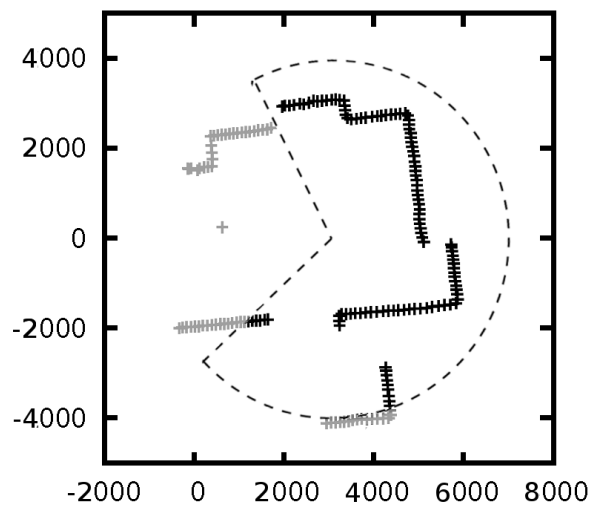


Bild 3.8: Filter: Sichtbarkeitsfilter auf Punkten. Die gestrichelten Linien beschreiben den maximal sichtbaren Bereich des Roboters. Die hell markierten Scanpunkte sind zu löschen.

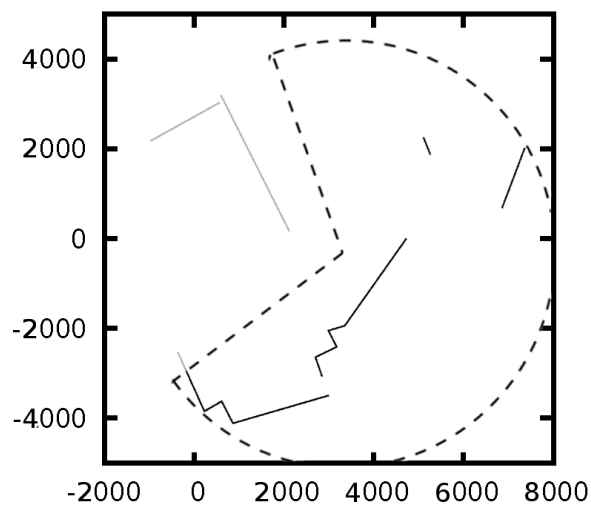


Bild 3.9: Filter: Sichtbarkeitsfilter auf Linien. Die gestrichelten Linien beschreiben den maximal sichtbaren Bereich des Roboters. Die hell markierten Linienfragmente sind aus dem Datensatz zu entfernen.

3.2.6 Reduzierung der Punkte

Da die Laufzeit der Scanmatching Verfahren hauptsächlich von der Anzahl der zu bildenden Punktkorrespondenzen abhängt, erscheint es sinnvoll die mögliche Menge der Korrespondenzpartner frühzeitig einzuschränken. Hierzu löschen wir diejenigen Punkte aus dem Laser-Scan, die keinen wichtigen Informationsgehalt haben. Wichtig ist jedoch im Gegenzug, dass die prägensten Informationen über die Struktur der Daten erhalten bleiben.

Unser Laserscanner liefert 682 Abstandsmessungen, die in regelmäßigen Winkelabständen erhoben werden. Befinden sich die Hindernisse nahe am Scanner, so liefert der Scanner dort viele Punkte auf engem Raum. Sind die Hindernisse jedoch weiter entfernt, so vergrößern sich logischerweise auch die Abstände zwischen den einzelnen Punkten. Mit unserem Reduktionsfilter möchten wir nun die Abstände zwischen den Scan-Punkten regelmäßiger gestalten. Zum einen haben solche nahen Regionen mit vielen Punkten bei der Transformationsbestimmung einen größeren Einfluss als die weiter entfernten, da von ihnen natürlich mehr Korrespondenzen in die zu minimierende Gleichung eingehen. Zum anderen berücksichtigen Verfahren wie der *Icp* während der Transformationsbestimmung auch einen Term, der bewirkt, dass die beiden Schwerpunkte der beiden Scans zu einander geschoben werden. Diese Eigenschaften fördert bei nicht vorbehandelten Daten das Verhalten, dass bevorzugt nur eine kleine oder gar keine Transformation bestimmt wird.

Gutmann [Gut00] stellt einen Reduktionsfilter vor, der genau diese Anforderungen umsetzt. In diesem Filter werden Punktwolken zu einem einzigen Punkt zusammen gefasst. Eine Punktwolke wird durch ihren Radius angegeben und man bestimmt sie wie im folgenden beschrieben: Jede Punktwolke besteht aus mindestens einem Punkt. Demnach wird der erste Punkt der ersten Wolke zugeordnet. Dann fügt man weitere Punkte hinzu, bis ein Punkt den Abstand des doppelten Radius zum ersten Punkt überschreitet. Geschieht dies, wird die Punktwolke geschlossen und mit dem einzelnen Punkt eine neue Gruppe eröffnet. Der Schwerpunkt der Gruppe ergibt dann den gemeinsamen Repräsentant in der Menge der verbleibenden Punkte.

Zu beachten ist, dass die Punkte in aufsteigender Winkelreihenfolge vorliegen müssen. Außerdem hat es sich bewährt, jeweils den ersten und den letzten Punkt des Scans eben-

falls zu übernehmen. Dies verhindert bei Wänden zum Beispiel, dass sie noch unnötig verkürzt werden.

Die Wirkungsweise lässt sich an Abbildung 3.10 nachvollziehen. Abbildung 3.10 a) zeigt die Original-Verteilung der Scan-Punkte, wohingegen Abbildung 3.10 b) die verbliebenen Punkte nach Anwendung des Reduktionsfilters darstellt.

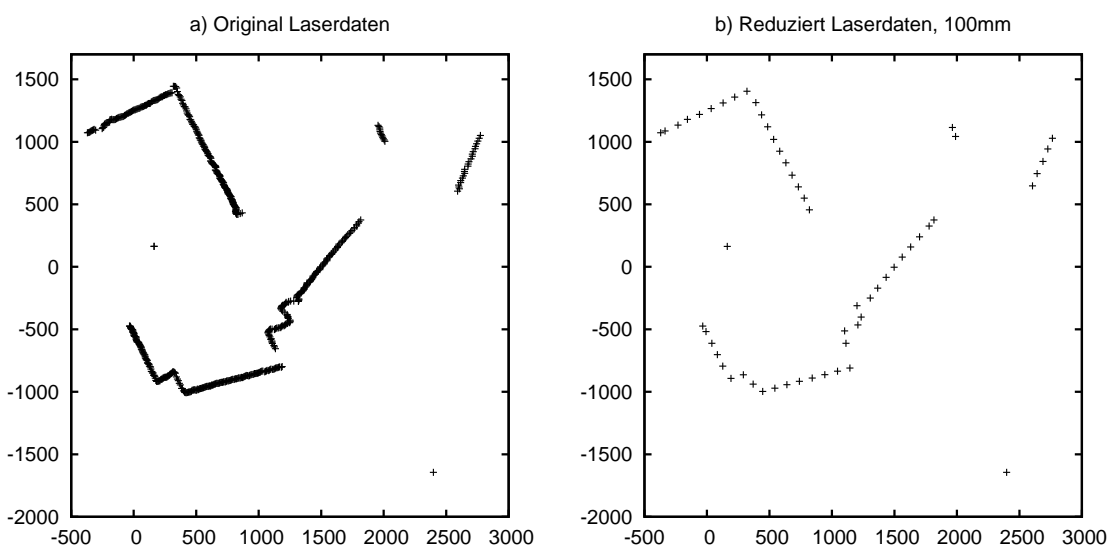


Bild 3.10: Filter: Reduzierung der Scan-Punkte. a) Original Laserdaten, 646 Scanpunkte; b) Reduzierte Laserdaten mit Filtergröße 100 mm, 58 Scanpunkte.

Der Reduktionsfilter wird durch folgenden Parameter gesteuert:

- *reducedPointDist* der ungefähre Abstand zweier aufeinanderfolgender Punkte.

3.3 Einbindung in das mobile System Robbie

3.3.1 Verwendete Daten

Laserdaten

Bei dem von Robbie eingesetzten Laserscanner handelt es sich um einen Hokuyo URG-04LX. Dieser liefert pro Messung 682 Entfernungsdaten in regelmäßigen Winkelabständen. Dabei liegen die Winkel in aufsteigender Reihenfolge vor und decken einen Winkelbereich von 240° ab. Aufgrund der Bauweise des Roboters gibt es im hinteren Bereich einen toten Winkel der verbleibenden 120° . Der Laserscanner ist auf Servo-Motoren gelagert, welche durch die Messungen von Lagesensoren gesteuert werden. Somit kann der Laserscanner auch auf unebenem Gelände horizontal gehalten werden. Die maximale Reichweite beträgt 5,60 Meter.

Odometriedaten

Die verwendete Pioneer Plattform des Roboters liefert automatisch Odometriedaten. Diese Daten werden aus den Radumdrehungen berechnet und sagen aus, wo der Roboter sich befinden *sollte*. Sie werden in globalen Millimeter-Einheiten angegeben und referenzieren auf den Startpunkt des Roboters. Auftretende Fehler werden somit aufaddiert. Als Stützung für das Scanmatching betrachten wir jedoch nur die lokale Bewegung zwischen dem Aufnahmeort des aktuellen Scans und dem des Referenz-Scans.

Odometriedaten sind relativ ungenau und werden daher nur als Stützung für den Scanmatcher oder den Partikelfilter verwendet. Typische Fehlerquellen im Rahmen des RoboCups Rescue sind durchdrehende oder abrutschende Räder, beziehungsweise Ungenauigkeiten der Messungen wegen unebenem Untergrund.

3.3.2 Koordinaten Systeme

Wenn man sich mit Laserdaten und Selbstlokalisierung beschäftigt, ist es stets wichtig sich zu beachten, zu welchem Koordinatensystem die vorliegenden Daten gehören. Die fünf wichtigsten Koordinatensysteme seien hier kurz vorgestellt. Sie wurden jedoch nicht im Rahmen dieser Studienarbeit eingeführt sondern beruhen auf früheren Arbeiten von Stefan Wirth. Lediglich die Einbeziehung der Klasse *Transformation* ist neu.

Ursprüngliches Laser Koordinatensystem

Bei dem ursprünglichen Laser Koordinatensystem liegen die Daten in dem Format vor, wie sie vom Laserscanner-Modul gesendet werden. Jede Messung wird durch zwei Werte repräsentiert, die zusammen den Punkt im 2D-Raum beschreiben. Der erste Wert gibt den Slot an, in dem der zweite Wert als Entfernung gemessen wurde. In den Laser-Rohdaten ist der Slot durch die Position im Entfernungsvektor gegeben. Er lässt sich mittels der Geräteeigenschaften der Laserscanners in Winkelinformation umrechnen.

In diesem Format werden die Scandaten als Eingabe zur Vorverarbeitung erwartet.

Laser Koordinatensystem

Bei dem Laser Koordinatensystem handelt es sich um ein kartesisches Koordinatensystem. Der Ursprung des Systems liegt auf der Position des Laserscanners. Die x-Achse zeigt nach vorne, die y-Achse nach links.

Dieses Koordinatensystem dient im allgemeinen nur zur Umrechnung zwischen dem ursprünglichen Laser Koordinatensystem und dem Roboter-Koordinatensystem. Jedoch wird es auch während der Laser-Vorverarbeitung verwendet.

Roboter Koordinatensystem

Das Roboter Koordinatensystem ist achsenparallel zum Laser-Koordinatensystem, jedoch befindet sich der Ursprung des Systems in der Robotermitte.

Auf Basis dieses Systems arbeitet jedes der Scanmatching Verfahren. Auch die berechnete Transformation liegt in diesem Koordinatensystem vor und muss vom *PartikelFilter* zurück in das *Welt Koordinatensystem* gerechnet werden.

Welt Koordinatensystem

Das Welt Koordinatensystem ist ebenfalls ein kartesisches System. Im Gegensatz zu den vorhergegangenen handelt es sich hierbei um ein globales System, da es nicht nur die aktuelle Umgebung des Roboters berücksichtigt. Der Ursprung des Systems liegt in der Position, in der der Roboter zu Beginn seiner Mission gestartet ist. Auch die Achsenausrichtung wird von der Startposition übernommen.

Die berechnete Roboterposition des Slam-Moduls ist in Weltkoordinaten und werden als solche an andere Module weiter gereicht. Auch die zur Orientierung gegebenen Odometriedaten sind in globalen Koordinaten gegeben. Diese müssen durch Fehlmessungen nicht mit den Weltkoordinaten übereinstimmen. Sie müssen jedoch vor dem Scanmatching vom *PartikelFilter* in das lokale *Roboter Koordinatensystem* übertragen werden.

Karten Koordinatensystem

Bei dem Karten Koordinatensystem handelt es sich um eine diskretisierte Repräsentationsform. Es entspricht der erstellten Karte durch das SlamModul. Der Ursprung liegt in der oberen linken Ecke, die x-Achse zeigt nach recht und die y-Achse nach unten. Die Ausrichtung des Koordinatensystems in der Welt ist wiederum abhängig von der Startposition.

Das Karten Koordinatensystem ist lediglich von Bedeutung, wenn man mit einer Position auf der Karte arbeiten möchte. Dies ist zur Zeit für den Scan-Matcher nicht von Nöten, eventuell ist dies jedoch für zukünftige Anwendungen erforderlich. Siehe hierzu Kapitel 5.

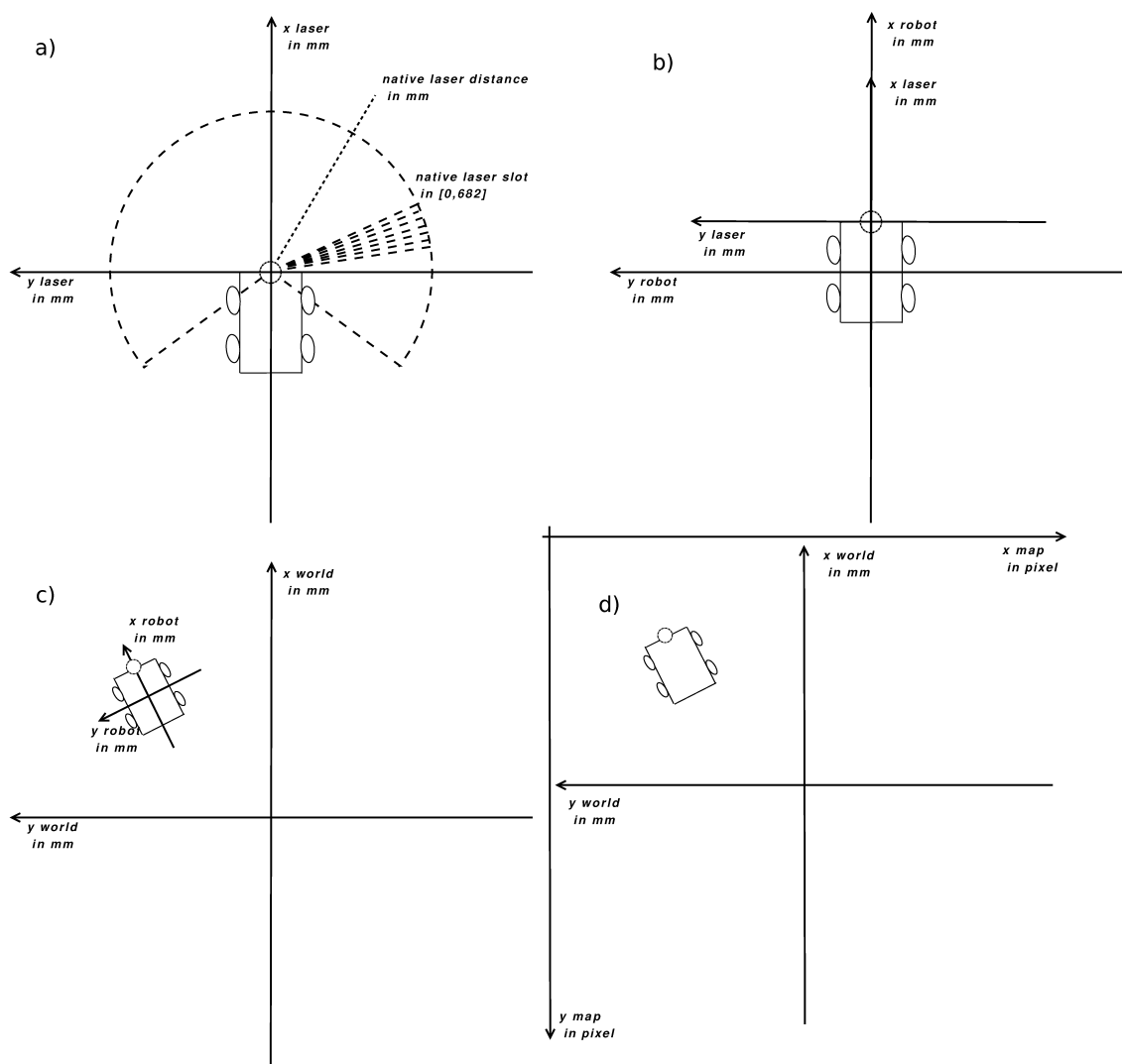


Bild 3.11: Verwendeten Koordinatensysteme: a) Ursprüngliches-Laser- und Laser-Koordinatensystem; b) Laser- und Roboter-Koordinatensystem; c) Roboter- und Welt-Koordinatensystem; d) Welt- und Karten-Koordinatensystem. Das rechteckige Symbol gibt jeweils eine mögliche Roboter-Position an. Der Kreis an der Oberseite gibt die Laser-Scanner Position an.

Transformation zwischen Koordinatensystemen

Um mit den Laserdaten effizient arbeiten zu können, muss man die Scanpunkte von einem Koordinatensystem in ein anderes transformieren können. Hierzu gibt es einen Koordinatenkonverter von Stephan Wirth. Für das Scanmatching benötigen wir jedoch zusätzlich noch die Transformation von Posen und Transformationen. Vor allem die Unterscheidung zwischen lokalen und globalen Systemen ist entscheidend für die Einbindung des Scanmatchers.

In Abbildung 3.11 sind die Beziehungen zwischen den Koordinatensystemen graphisch dargestellt.

3.3.3 Einbindung in Kartenerstellung und Selbstlokalisierung

Der Hauptverwendungszweck der Scanmatching Verfahren ist im Rahmen der Kartenerstellung und Selbstlokalisierung. Im Robbie-System ist dies durch ein *Simultaneous-Localisation-And-Mapping (SLAM)* Verfahren gelöst, welches durch einen Partikelfilter realisiert wird. Diese beiden Techniken sollen daher im folgenden kurz beschrieben werden.

SLAM

Das Hauptproblem bei der Kartenerstellung ist, festzustellen wo auf der Karte der Roboter sich aktuell befindet. Um sich jedoch auf einer Karte lokalisieren zu können, benötigt man erst einmal eine Karte. Dieser triviale Umstand führt dazu, dass Lokalisation und Kartenerstellung untrennbar miteinander verbunden sind. Als *SLAM (Simultaneous Localisation And Mapping)* Verfahren bezeichnet man diejenigen Algorithmen, die anstreben Kartenerstellung und Lokalisation gleichzeitig zu lösen. Die Slam Verfahren sind die aktuell am häufigsten verwendeten Verfahren zur Kartenerstellung.

Partikelfilter

Der Schwerpunkt des Partikelfilters liegt auf der Lokalisation des Roboters in der bereits initialisierten Karte. Der Partikelfilter streut mehrere Partikel im Raum, welche alle eine mögliche Pose des Roboters darstellen. Diese Posen werden in jedem Mapping-Schritt analog zu den gegebenen Odometrie-Daten verschoben. Desweiteren wird ein Rauschen auf die einzelnen Posen gelegt, welches die Fehler der Odometrie-Daten ausgleichen und eine Streuung der Partikel gewährleisten soll. Auf die Korrektheit der Odometrie-Daten kann man sich leider nicht verlassen, da durch Ansteigungen im Boden, rutschigen Untergrund oder gar nur unterschiedlichem Luftdruck in den Reifen fehlerhafte Messungen zustande kommen. Zudem addieren sich die Fehler mit der Zeit auf und zerstören somit die globale Korrektheit der Karte.

Nach ihrer Verschiebung werden die einzelnen Partikel durch einen Abgleich mit den aktuellen Laserdaten gewichtet. Dies geschieht dadurch, dass man die Laserstrahlen von der Partikelposition ausgehend in das Karten Koordinatensystem übersetzt und dort mit der bereits berechneten Karte vergleicht. Die Partikel mit vielen Übereinstimmungen erhalten eine hohe Gewichtung, sowie die Partikel mit wenig Übereinstimmungen eine geringe erhalten.

Die Aktualisierung der Karte wird dadurch vorgenommen, dass an der Position des stärksten Partikels die Laserdaten in die Karte eingetragen werden. Außerdem lässt man mit jedem Durchlauf die schlechtesten Partikel aussterben und generiert anstelle dessen neue aus den stärksten Partikeln.

Die Abbildungen 3.12 und 3.13 zeigen das Vorgehen etwas detaillierter. Die Bezeichner in Abbildung 3.13 sind auf englisch, da sie sich auf die original Bezeichner des Verfahrens beziehen. Für ein Verständnis im Detail empfiehlt sich weitere Literatur, wie zum Beispiel [IB98].

Einbindung des Scanmatchings

Es gibt unterschiedliche Möglichkeiten ein Scanmatching in die Kartenerstellung zu integrieren. Eine naheliegende Vorgehensweise ist die Optimierung der Partikelpositionen

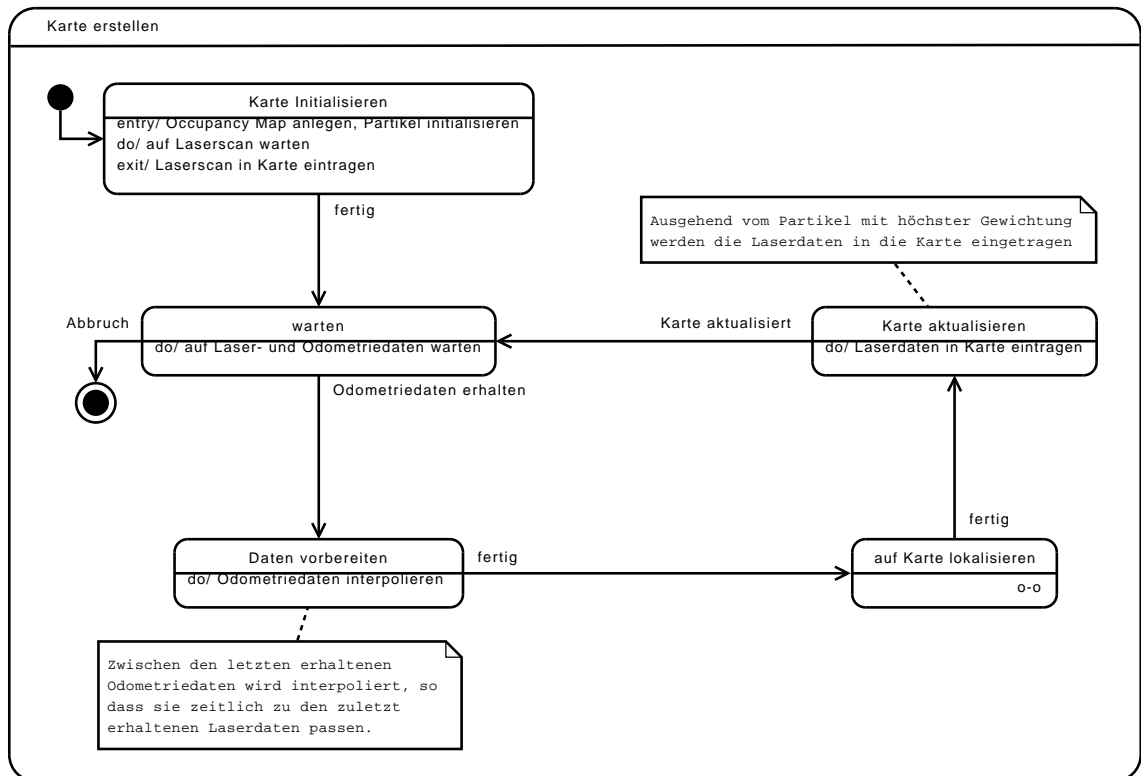


Bild 3.12: Statechart: allgemeiner Ablauf der Kartenerstellung mittels *SLAM* Verfahren. Zur Verfeinerung von *auf Karte Lokalisieren* siehe 3.13.

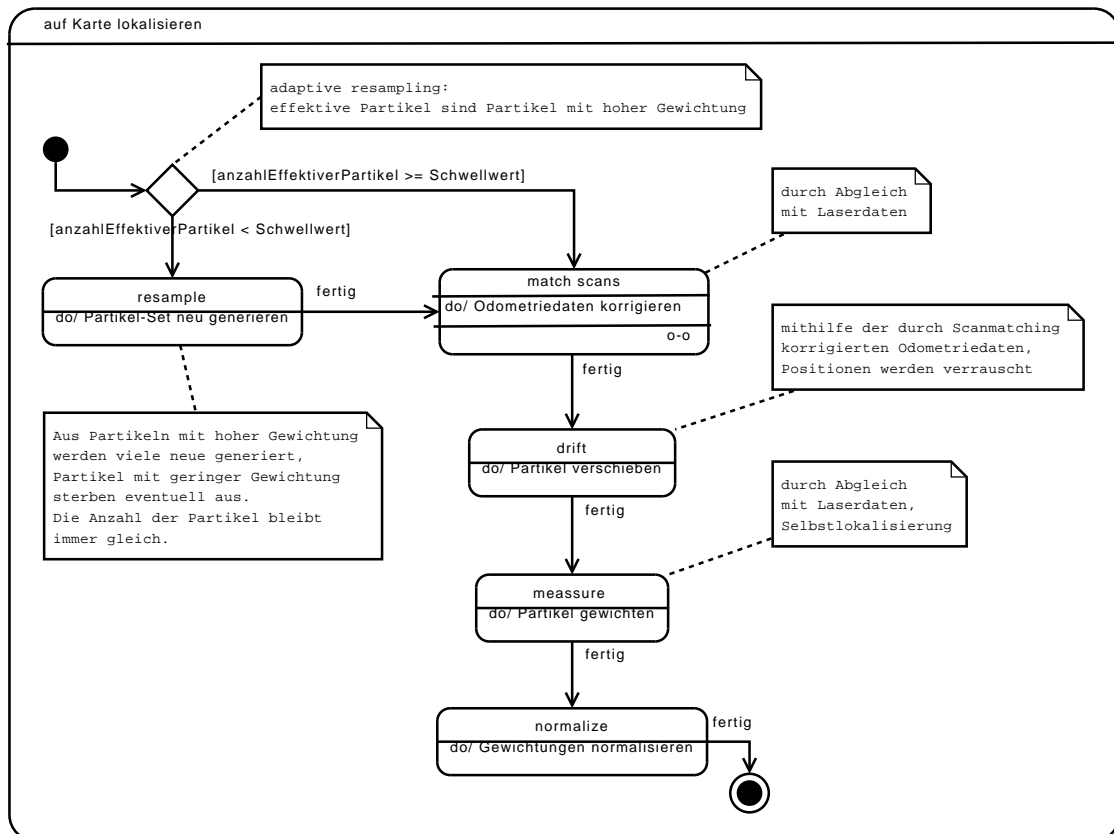


Bild 3.13: Statechart: einzelner Schritt zur Selbstlokalisierung im Partikelfilter.

mittels Scanmatching. Hierbei würde man die aktuellen Laserdaten gegen die gegebene Karte matchen.

Während dieser Studienarbeit wurde sich jedoch auf die Korrektur der Odometriedaten beschränkt. Abbildung 3.13 zeigt den Ablauf der Lokalisierung auf einer Karte mit integrierter Korrektur der Odometriedaten.

In Abschnitt 3.3.1 wurden bereits die Eigenschaften und möglichen Fehlerquellen der Daten beschrieben. Um etwaige Fehler auszugleichen, können die lokal ermittelten Odometriedaten mittels Scanmatching korrigiert werden. Hierbei versucht man die aktuellen Laserdaten auf die Referenzdaten des vorherigen Mapping-Schrittes abzubilden. Ziel ist es, die Transformation zwischen dem Aufnahmeort der Referenzdaten und der aktuellen Roboterpose zu bestimmen. Die aus den Odometriedaten ermittelte Transformation stellt die initiale Schätzung für das Scanmatching dar.

Das Vorgehen während des Scanmatchings kann mithilfe des Aktivitätsdiagrammes 3.14 nachvollzogen werden.

Mit dem Ergebnis des Scanmatchings wird der Wert der Referenz-Odometriedaten korrigiert. Daher fließt das Ergebnis des Scanmatchings nur indirekt in die weiteren Berechnungen mit ein. Das Scanmatching bildet somit eine zusätzliche, aber nicht notwendige Stützung der Odometriedaten. Über Konfigurationseinstellungen kann das Verfahren gezielt aktiviert oder deaktiviert werden. Die Referenz-Odometriedaten werden während des aktuellen Mapping-Schrittes nur zur Berechnung der neuen Partikelpositionen verwendet und anschließend verworfen. Der Wirkungskreis des Scanmatchings bleibt somit auf die aktuellen Berechnungen des Partikelfilters beschränkt.

Abbildung 3.15 zeigt die Struktur der am Scanmatching beteiligten Klassen. Der Partikelfilter verfügt über seinen eigenen ScanMatcher, wessen Verfahrensparameter er mittels Attributen bestimmen kann. Zur Durchführung des Scanmatchings übergibt er die aktuellen Daten mit einer Referenz auf seinen ScanMatcher an die Utils-Klasse. Diese führt die Datenvorverarbeitung und anschließend das Scanmatching aus. Die Wahl des Verfahrens und der Verfahrensparameter sind somit dem Partikelfilter überlassen, ohne unnötige Intelligenz des Scanmatchers dorthin auszulagern.

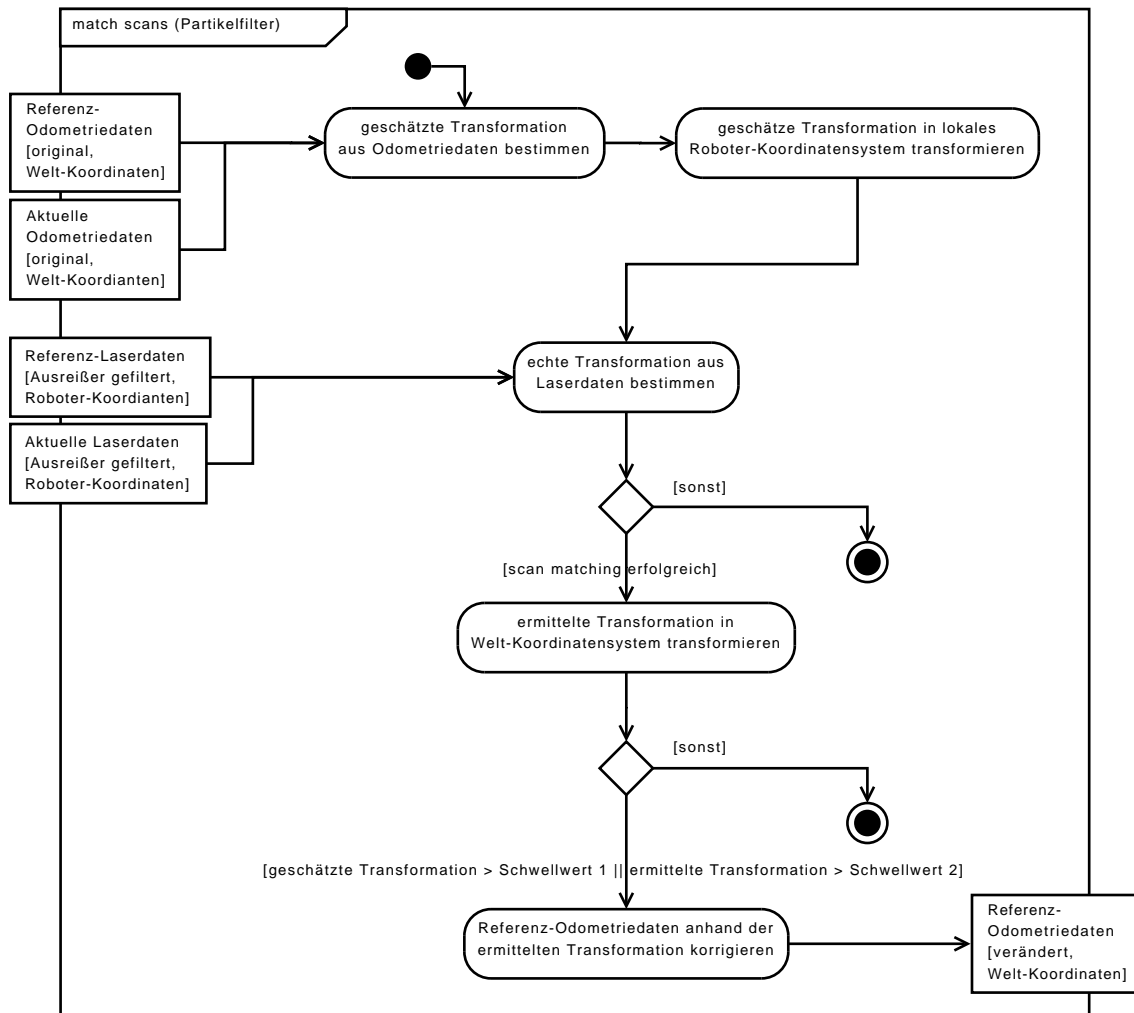


Bild 3.14: Statechart: Einbindung des Scanmatchings im Partikelfilter.

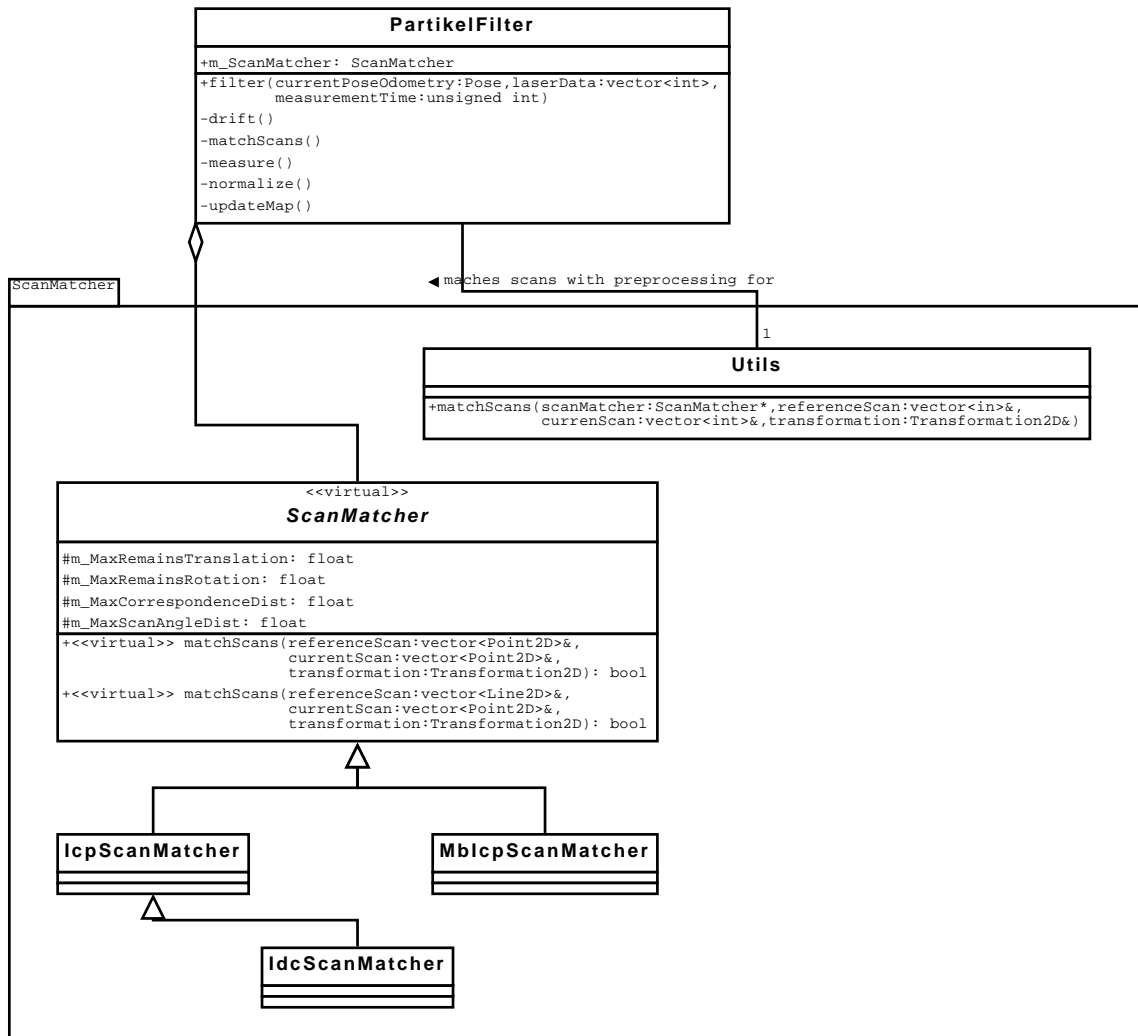


Bild 3.15: Klassendiagramm: Scan-Matcher und Partikelfilter

Mögliche Probleme und deren Behandlung Sollte keine geeignete Transformation zwischen den Laserscans gefunden werden, liefert das Verfahren einen *false*-Wert zurück. In diesem Fall werden die Odometriedaten nicht verändert und es wird mit den original Daten weitergearbeitet.

Ein weiteres Problem kann auftreten, wenn der Roboter längere Zeit still steht. Durch Rauschen der Laserdaten und kleine Ungenauigkeiten tendiert ein Scanmatching Verfahren dazu, trotz Stillstand kleine Transformationen zu berechnen. Daher ist es sinnvoll, bei einer Odometrie-Differenz von $q_0 = (0, 0, 0)$ das Resultat des Scanmatchers zu ignorieren, sollte keine deutliche Bewegung ermittelt werden. Ansonsten kann es passieren, dass die Karte langsam zerbricht, da auch der Partikelfilter diesen Fehler auf Dauer nicht auffängt. Das Resampling des Partikelfilters ist aus ähnlichen Gründen bei zu geringer Bewegungen deaktiviert.

Kapitel 4

Experimente und Ergebnisse

In diesem Kapitel werden zunächst die optimalen Werte für die wichtigsten Parameter bestimmt. Im Anschluss werden die kompletten Verfahren gegeneinander getestet und auf Schwachstellen und Vorzüge überprüft. Daraufhin wird die Nützlichkeit des Scanmatchers durch Einbindung in die Kartenerstellung evaluiert.

Da die Evaluationsergebnisse stark von der Wahl der Parameter abhängen, werden diese für jeden Test gesondert angegeben. In der Tabelle 4.1 sind die wichtigsten Parameter aufgestellt und kurz erklärt.

Bei der Evaluation gibt es zwei generell verschiedene Vorgehensweisen:

- die Evaluation auf ausgewählten Laserdaten und
- die Evaluation auf ganzen Karten.

Evaluation auf ausgewählten Laserdaten

Handelt es sich um eine Evaluation auf einzelnen Laserdaten, so werden standardmäßig die in Abbildung 4 vorgestellten Datensätze verwendet. Sie spiegeln je nach Rotation und Translation Situationen mit unterschiedlichen Schwierigkeitsgraden dar. Außerdem sind die Referenz-Daten und die darauf zu matchenden aktuellen Daten nicht identisch, sondern sind um eine halbe Sekunde versetzt aufgenommen worden. Dies entspricht dem

Parameter	Bedeutung
<i>maxRemainingTranslation</i>	Verbleibende Translation , bei der der Algorithmus als konvergiert betrachtet und abgebrochen wird. [<i>in mm</i>]
<i>maxRemainingRotation</i>	Verbleibende Rotation, bei der der Algorithmus als konvergiert betrachtet und abgebrochen wird. [<i>in Radianen</i>]
<i>maxCorrespondenceDist</i>	Maximaler Abstand zwischen zulässigen Punktkorrespondenzen. [<i>in mm</i>]
<i>maxScanAngleDist</i>	Maximaler Winkel-Abstand zwischen zulässigen Punktkorrespondenzen. [<i>in Grad</i>]
<i>minReducedPointDist</i>	Minimaler Abstand zweier Punkte nach Anwendung des Reduktionsfilters. [<i>in mm</i>]
<i>maxInterpolationDist</i>	Maximaler Abstand zweier Punkte zwischen denen interpoliert wird. [<i>in mm</i>]
<i>metricFactor</i>	Beeinflusst das Verhalten des MbIcps bei der Korrespondenzbestimmung.
<i>deletedCorrespondences</i>	Prozent der gelöschten Punktkorrespondenzen bei Ausreisser Eliminierung. Gelöscht werden die Korrespondenzen mit dem größten Abstand zueinander.

Tabelle 4.1: Parameterübersicht

natürlichen Rauschen, welches die Scanmatcher erwarten müssen.

Die Tests umfassen stets:

- die Rotation von -45° bis 45° Grad
- die Translation von -300 mm bis 300 mm
- die gleichzeitige Rotation von -200 mm bis 200 mm und 45° bis 45° Grad in unterschiedlichen Kombinationen.

Dies wird als ausreichend betrachtet, da der Haupteinsatzzweck des Scanmatchers zur Ergänzung der Odometriedaten gedacht ist und daher nicht mit übermäßig großen Transformationen konfrontiert wird. Eine Rotation von mehr als 45° Grad liefert im Zusammenhang mit Scanmatchern meist kein verlässliches Ergebnis, da es sich in den meisten Fällen um rechtwinklige Umgebungen handelt.

Als Kriterien zur Bewertung gelten:

- die Anzahl der aufgetretenen Fehlerfälle,
- die Anzahl der benötigten Iterationen und
- die Abweichung der bestimmten Transformation von der erwarteten Transformation.

Ein Fehlerfall liegt dann vor, wenn das Verfahren nicht konvergiert oder das gefundene Ergebnis als zu schlecht verworfen wird. Für die genauen Kriterien siehe Kapitel 3.1.2.

Evaluation auf ganzen Karten

Findet die Evaluation auf ganzen Karten statt, so erfolgt dies durch das Abspielen von Logfiles der original RoboCup-Arenen der GermanOpen 2008 in Hannover und der Weltmeisterschaft 2008 in Suzhou. Beispiel-Karten sind in Abbildung 4 zu sehen.

Als Kriterien zur Bewertung sind hierbei ebenfalls:

- die Anzahl der aufgetretenen Fehlerfälle,

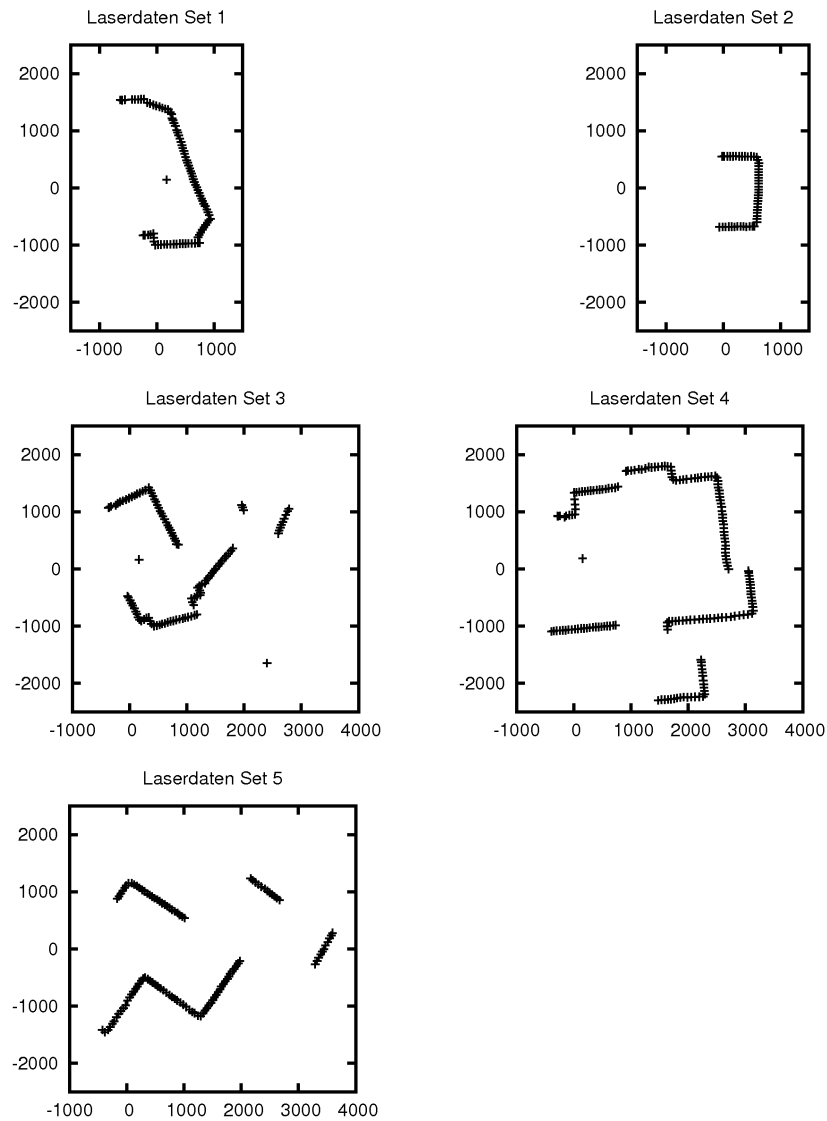


Bild 4.1: Testdaten für die Evaluation auf ausgewählten Datensätzen. Zur Übersichtlichkeit wurde auf die hier dargestellten Laserdaten ein Reduktionsfilter mit der Maskengröße 50 angewendet.

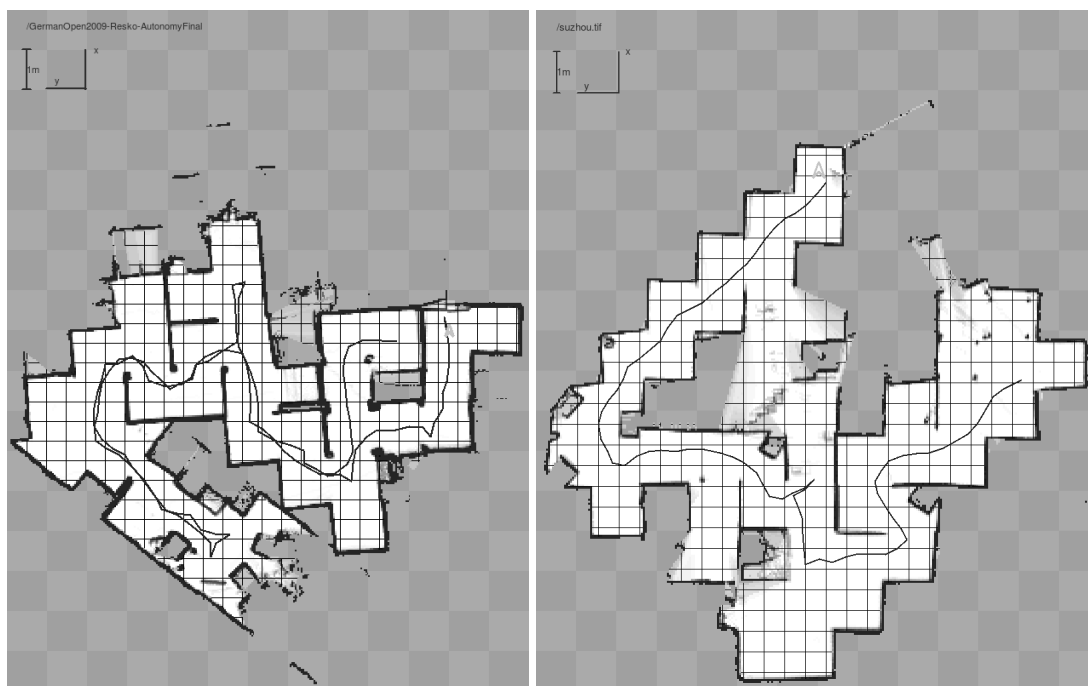


Bild 4.2: Testdaten zur Evaluation auf ganzen Karten. Weiße Fläche stellt freies Gelände dar, schwarz sind die Hindernisse, bei grauen Flächen ist der Roboter unsicher. Der Strich in der Gangmitte verzeichnen den vom Roboter eingeschlagenen Pfad. Der hellgraue Pfeil markiert die Startposition.

- die Anzahl der durchschnittlich benötigten Iterationen und
- die Kartenqualität.

Die Kartenqualität wird automatisch nach einem Verfahren von Dipl. Inf. Johannes Pelenz berechnet. Das Verfahren bezieht sich auf die Schärfe der Karte als Qualitätsmaß. Verwaschene und somit unsichere Stellen gehen negativ in die Bewertung ein und verschlechtern das Resultat der Kartenqualität. Dennoch ist eine visuelle Überprüfung der erzeugten Karten anzuraten.

4.1 Parameterjustierung

Entscheidend für den Erfolg des Scanmatchings ist die richtige Wahl aller Parameter. Die wichtigsten seien hier vorgestellt und ein möglichst optimaler Wert berechnet.

4.1.1 Konvergenzkriterien

Versuchsaufbau

Alle drei Scanmatching Verfahren beenden ihre Berechnungen, wenn die aktuell ermittelten Transformationsverbesserungen einen festgelegten Schwellwert unterschreiten. Diese Werte werden für Rotation und Translation gesondert abgefragt und durch die Parameter *maxRemainingTranslation* und *maxRemainingRotation* festgelegt.

Versuchsdurchführung

Der Test wird mittels der zuvor vorgestellten Standard-Testszenarien 4 durchgeführt. In Tabelle 4.4 sind die verwendeten Programmparameter aufgelistet. Im Anschluss wird das Ergebnis mittels eines Tests auf der ganzen Karte verifiziert.

Parameter	1. Parameter-Set	2. Parameter-Set	3. Parameter-Set
<i>maxRemainingTranslation</i>	0.1	1.0	1.0
<i>maxRemainingRotation</i>	0.0001	0.001	0.01
<i>maxCorrespondenceDist</i>	800.0	800.0	800.0
<i>maxScanAngleDist</i>	80.0	80.0	80.0
<i>minReducedPointDist</i>	50.0	50.0	50.0
<i>maxInterpolationDist</i>	150.0	150.0	150.0
<i>metricFactor</i>	1500	1500	1500

Tabelle 4.2: Parameterbelegung zur Evaluation der Konvergenzkriterien

Ergebnisse

Evaluation auf ausgewählten Laserdaten Die Qualität der Resultate des ersten und zweiten Parametersets unterscheiden sich nur geringfügig. Die Qualität des dritten Parametersets fällt gegen die ersten beiden jedoch deutlich ab, wie in Abbildung 4.3 zu sehen ist. In der Bestimmung der Rotation tritt ein durchschnittlicher Fehler von $0,5^\circ$ auf. Dies ist zwar immer noch nicht viel, jedoch fallen die Rotationsfehler in der Kartenerstellung am gravierendsten auf.

Evaluation auf ganzen Karten Die Evaluation auf ganzen Karten stützt diese Resultate. Die ersten beiden Parametersets unterscheiden sich nur geringfügig, was die Kartenqualität oder die Anzahl der auftretenden Fehlerfälle betrifft. Die Anzahl der Iterationen reduziert sich von Parameterset eins zu Parameterset zwei jedoch um 60 %. In den Diagrammen 4.4 ist zu erkennen, dass die Verfahren unterschiedlich gut mit den verschiedenen Konvergenzkriterien umgehen können. Der *Idc* weist eine deutliche Vorliebe für den mittleren Genauigkeitsgrad auf. Der *Mblcp* hingegen arbeitet mit allen Konvergenzkriterien mit gleichbleibender Genauigkeit. Lediglich die Anzahl der benötigten Iterationen nimmt ab. Er scheint am robustesten zu arbeiten.

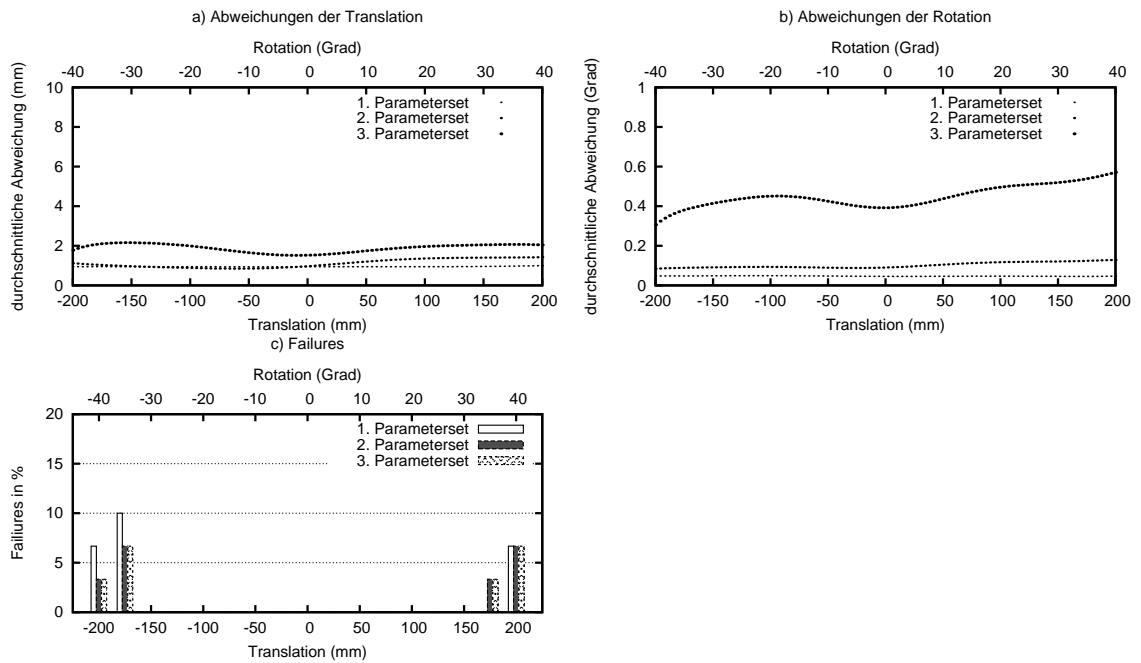


Bild 4.3: Abweichungen der Translationen und auftretende Fehlerfälle für unterschiedliche Konvergenzkriterien. Die dargestellten Ergebnisse umfassen gleichzeitige Rotation und Translation auf einzelnen Laserdaten für alle Verfahren. a) Abweichungen in der Rotation; b) Abweichungen in der Translation; c) Fehlerfälle in Prozent.

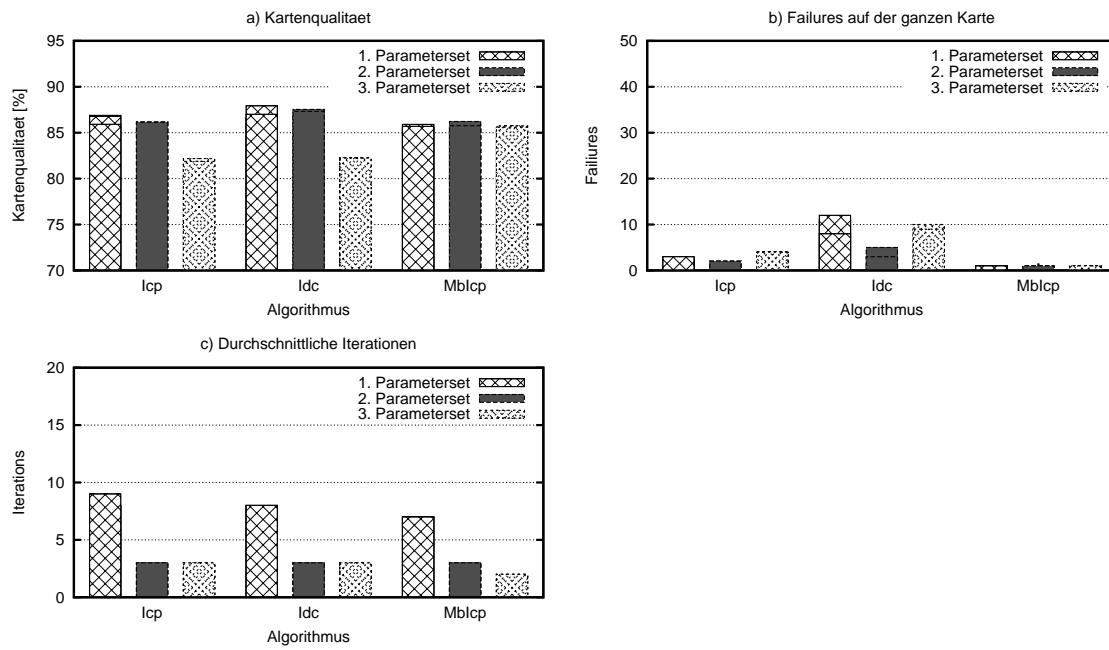


Bild 4.4: Resultate für unterschiedliche Konvergenzkriterien auf ganzen Karten. a) Resultat der automatischen Kartenbewertung; b) Anzahl aufgetretene Fehlerfälle; c) durchschnittlich benötigte Iterationen.

Parameter	Wert
<i>maxRemainingTranslation</i>	1.0
<i>maxRemainingRotation</i>	0.001
<i>maxCorrespondenceDist</i>	800.0
<i>maxScanAngleDist</i>	80.0
<i>minReducedPointDist</i>	50.0
<i>maxInterpolationDist</i>	150.0
<i>metricFactor</i>	200-4000
<i>deletedCorrespondences</i>	0%

Tabelle 4.3: Parameterbelegung während der Bestimmung des metrischen Faktors

4.1.2 Metrischer Faktor

Versuchsaufbau

Der *Metrische Faktor* bestimmt die Eigenschaften der *metrischen Distanz* des MbIcps und beeinflusst somit maßgeblich sein Verhalten bei der Korrespondenzbestimmung. Ein kleiner Wert bevorzugt die Rotation, wohingegen ein großer Wert das Verhalten des Verfahrens dem des *Icp* annähert. Ziel dieses Tests ist es, eine optimale Belegung für diesen Parameter zu finden.

Versuchsdurchführung

Der Test wird mittels der zuvor vorgestellten Standard-Testszenarien durchgeführt. In Tabelle 4.3 sind die verwendeten Programmparameter aufgelistet.

Ergebnisse

Die Resultate der Testläufe sind in den Diagrammen 4.6 und 4.5 zusammengefasst.

Das Diagramm 4.5 a) zeigt erfreulicher Weise, dass die Anzahl der auftretenden Fehlerfälle durch die Wahl eines Faktors >1800 auf 0 gesenkt werden kann. Hierbei überrascht

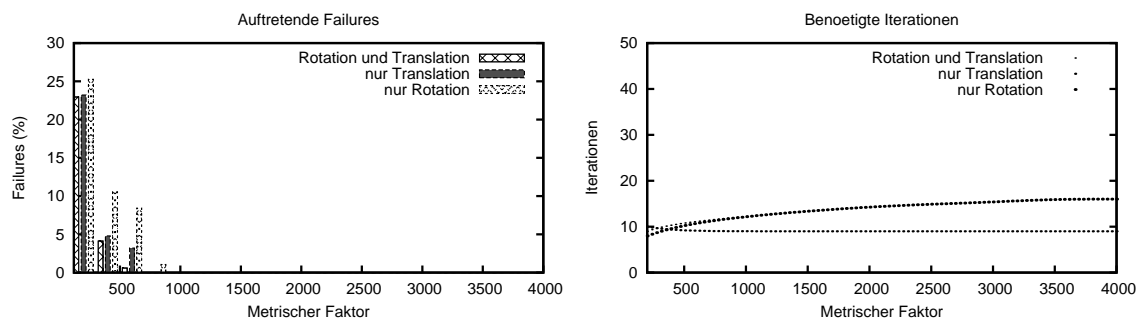


Bild 4.5: Die auftretenden Fehlerfälle in Prozent und die benötigten Iterationen bei wechselndem metrischen Faktor.

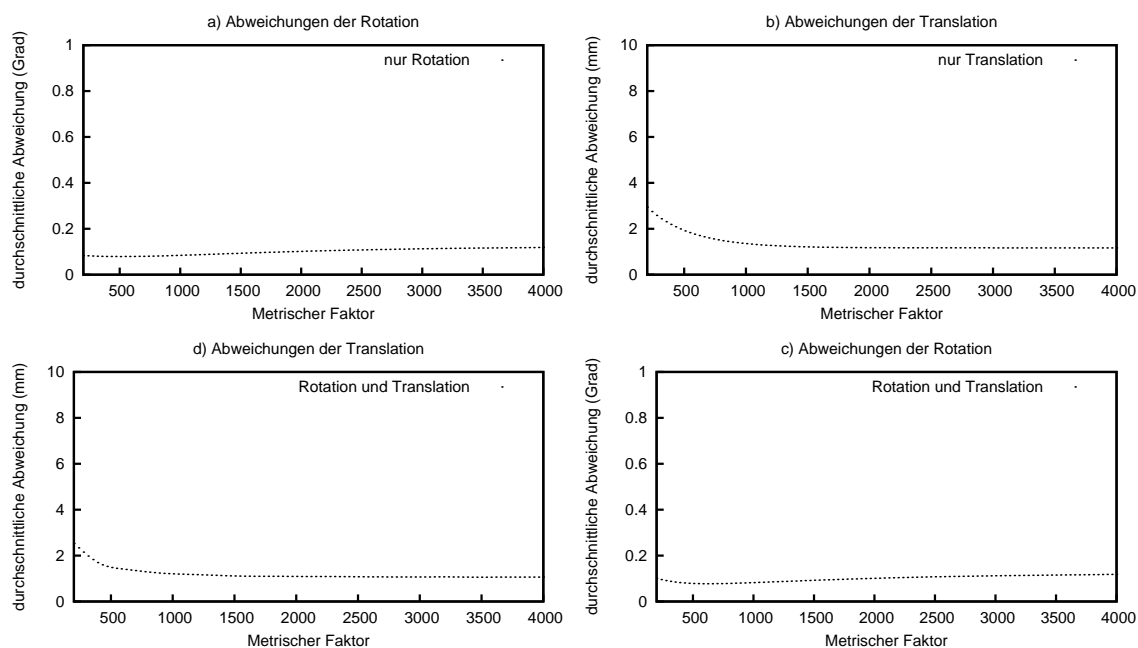


Bild 4.6: Abweichungen von der gesuchten Transformation für unterschiedliche Wertebereiche des metrischen Faktors.

es jedoch, dass eine starke Rotation scheinbar einen größeren Faktor bevorzugt. Da es sich jedoch um geringe Abweichungen handelt, können wir sie ignorieren.

Nach Diagramm 4.5 b) steigt die Anzahl der benötigten Iterationen mit der Größe des Faktors. Ab einem Wert von 1500 ist dieser Anstieg jedoch nur noch sehr schwach und pendelt sich bei durchschnittlich 15-18 Iterationen ein.

Die Ergebnisse aus Diagramm 4.5 unterstützen die bisherigen. Die Abweichungen der berechneten Transformationen von den gesuchten ist trotz wechselndem metrischen Faktors stabil gering. Nur bei Werten < 1500 treten größere Abweichungen auf.

Um die Vorteile des $MbIcp$ s zum Icp zu nutzen, ist es sinnvoll den metrischen Faktor nicht zu groß zu wählen. Daher bietet sich eine Belegung des Parameters mit 1800-2000 an.

4.1.3 Eliminierung falscher Punkt-Korrespondenzen

Versuchsaufbau

Bei der Zuordnung der Korrespondenzen können natürlich auch falsche Korrespondenzen bestimmt werden. Um die Qualität der Verfahren zu heben, können diese ausfindig gemacht und gelöscht werden. In Kapitel 3.1.5 wurde beschrieben, wie falsche Korrespondenzpunkte anhand ihrer Entfernung voneinander zu identifizieren sind. Im Folgenden wird der Erfolg und die Effizienz dieses Verfahrens getestet.

Zu beachten ist die unterschiedliche Einbindung des Ausreißer-Filters. Icp und $MbIcp$ wenden ihn Mangels anderer Alternativen auf alle Punkt-Korrespondenzen an. Im Idc wird der Filter jedoch nur verwendet, um die falschen Korrespondenzen der Translation zu eliminieren. Die Korrespondenzen zur Bestimmung der Rotation bleiben unberührt, da dort eine große Entfernung naturgemäß keine schlechte Korrespondenz bezeichnen muss.

Versuchsdurchführung

Der Test wird mittels der in der Einleitung vorgestellten Standard-Testszenarien durchgeführt. In Tabelle ?? sind die verwendeten Programmparameter aufgelistet.

Parameter	Wert
<i>maxRemainingTranslation</i>	1.0
<i>maxRemainingRotation</i>	0.001
<i>maxCorrespondenceDist</i>	800.0
<i>maxScanAngleDist</i>	80.0
<i>minReducedPointDist</i>	50.0
<i>maxInterpolationDist</i>	150.0
<i>metricFactor</i>	1500
<i>deletedCorrespondences</i>	0% bzw. 20%

Tabelle 4.4: Parameterbelegung zur Evaluation der Ausreißer Eliminierung

Ergebnisse

Die Auswirkung des Filters ist für die einzelnen Verfahren sehr unterschiedlich. Für den *Idc* sind durchaus positive Effekte bemerkbar, wohingegen *Icp* und *MbIcp* deutlich schlechter unter Einsatz des Filters arbeiten.

Eine erste Aussage lässt sich bereits nach Betrachten der Fehlerfall-Entwicklung in Abbildung 4.1.3 treffen. Für *Icp* (Diagramme a) bis d)) und *MbIcp* (Diagramme k)-l)) treten bei Anwendung des Filters deutlich mehr Fehlerfälle auf, als dies ohne der Fall ist. Vor allem bei großen Transformationen scheint der Filter ein Hindernis darzustellen. Der *Idc* hingegen weist bereits ohne Filter überdurchschnittlich viele Fehlerfälle auf. In den Diagrammen 4.1.3 e)-i) erkennt man dann deutlich, dass die Überprüfung der Korrespondenzpunkte durch den Filter die Qualität des Verfahrens hebt. Diese Erkenntnis wird gestützt durch die Qualität der berechneten Transformationen, siehe hierzu die Übersicht 4.1.3.

Auch bei der Anzahl der benötigten Iterationen nach Abbildung 4.1.3 wird dieser Eindruck gestärkt. Während *Icp* und *MbIcp* durch den Filter über 50% mehr Durchläufe benötigen, bleibt die Anzahl der Iterationen für den *Idc* gleich. Man muss jedoch berücksichtigen, dass der Filter einen Mehraufwand der Aufwandsklasse $\mathcal{O}(n \cdot \log(n))$ mit sich bringt, wobei n die Anzahl der Punkte ist.

Aufgrund der Qualitätsgewinnung für den *Idc* bietet es sich an, die *falschen Korrespondenzen* dort zu entfernen. Für die anderen beiden Verfahren macht dieser Eingriff keinen

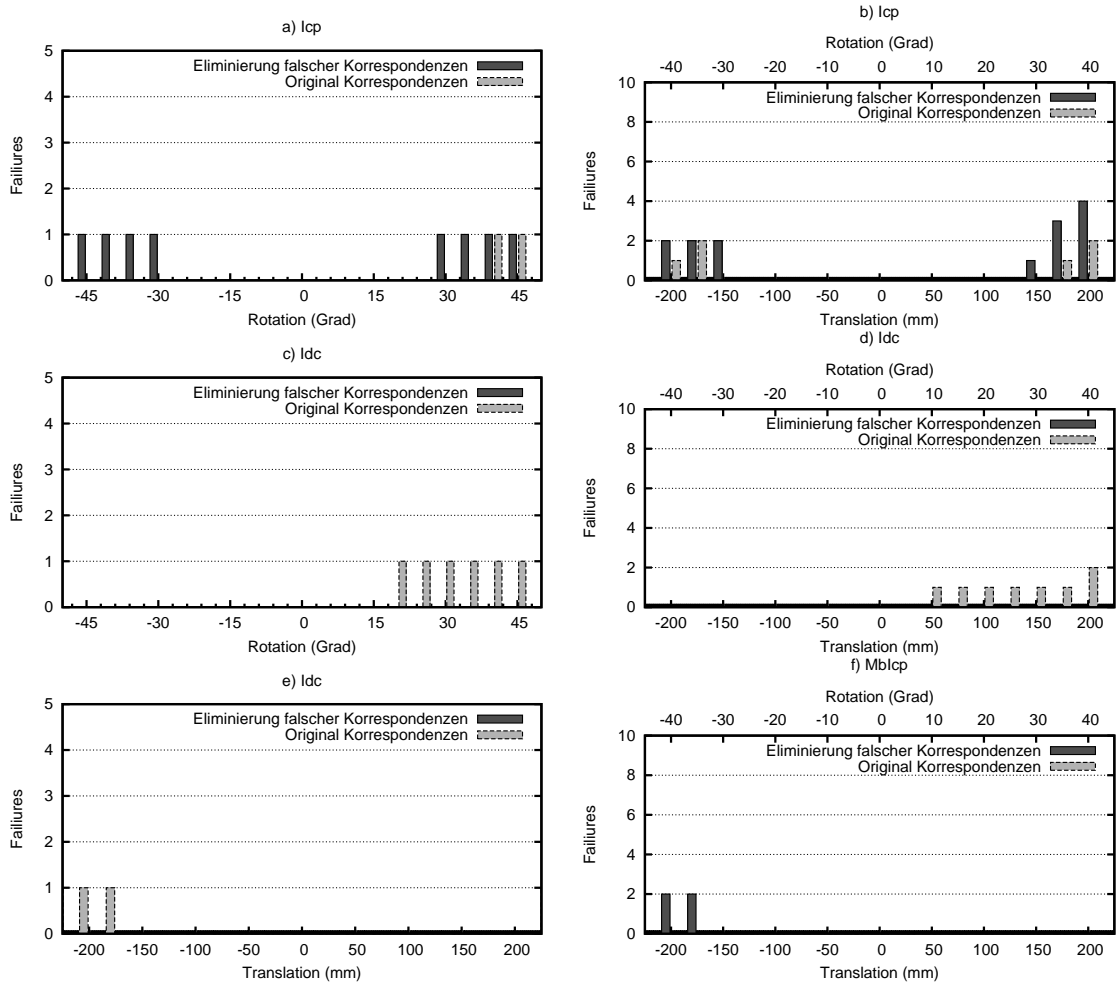


Bild 4.7: Eliminierung falscher Punktkorrespondenzen: auftretende Fehlerfälle bei der Transformationsbestimmung. Auf der linken Seite sind die Fehlerfälle bei ausschließlicher Rotation verzeichnet, auf der rechten Seite die der gleichzeitigen Rotation und Translation. Bei ausschließlicher Transformation traten keine Fehlerfälle auf.

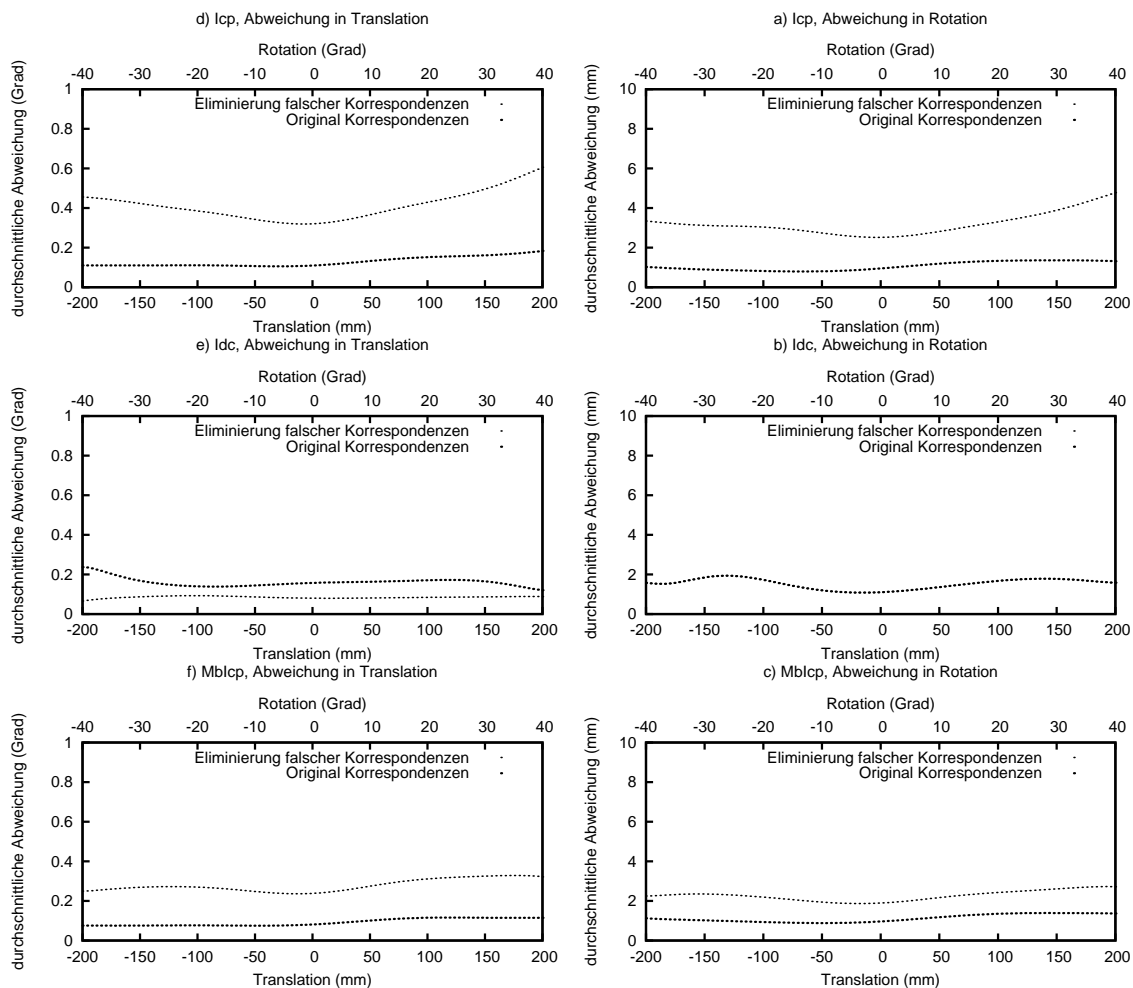


Bild 4.8: Eliminierung falscher Punktkorrespondenzen: Abweichungen von der gesuchten Transformation bei gleichzeitiger Rotation und Translation.

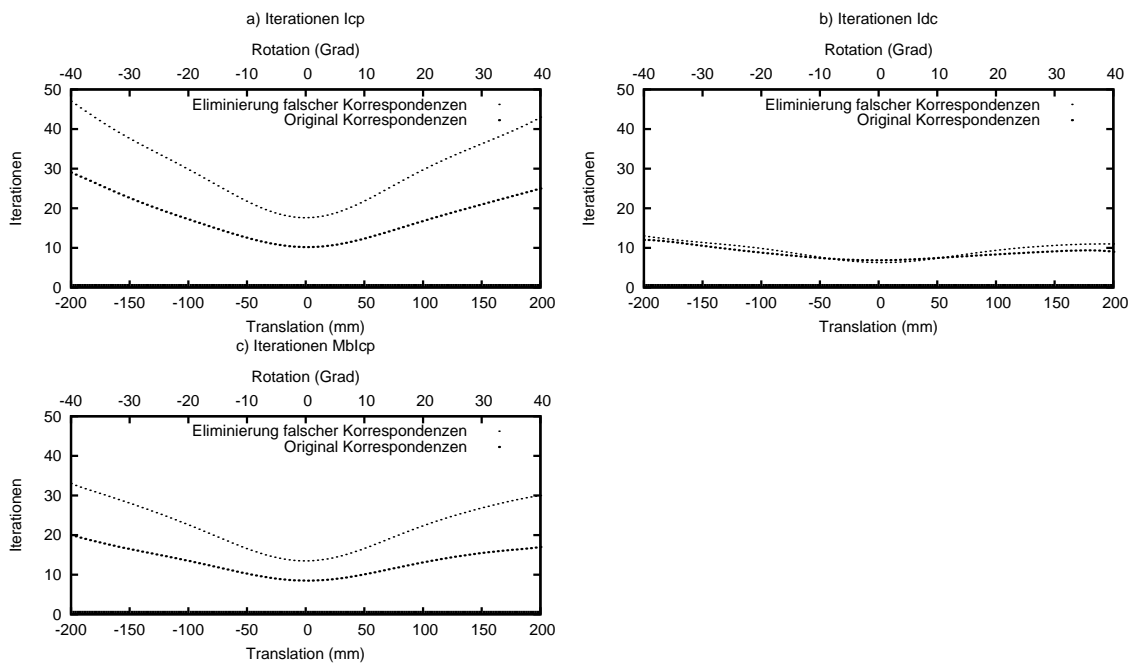


Bild 4.9: Eliminierung falscher Punktkorrespondenzen: benötigte Iterationen bis zur Konvergenz bei gleichzeitiger Rotation und Translation.

Sinn.

4.1.4 Maskengröße des Reduktionsfilters

Versuchsaufbau

Um Rechenzeit zu sparen, wird vor Beginn des Scanmatchers ein Reduktionsfilter eingesetzt. Dieser reduziert die Anzahl der Scanpunkte. Es ist zu erwarten, dass die Wahl einer großen Filtermaske das Verfahren sehr beschleunigt, aber auch zu großen Ungenauigkeiten führen kann. Der beste Kompromiss zwischen Qualität und Geschwindigkeit soll durch diesen Test ermittelt werden.

Versuchsdurchführung

Der Evaluation wird sowohl auf ausgewählte Laserdaten als auch auf der ganzen Karte durchgeführt. Die verwendeten Programmparameter sind hier nicht von Bedeutung und entsprechen den Standardparametern der vorhergegangenen Tests.

Ergebnisse

Evaluation auf ausgewählten Laserdaten Bei der Evaluation auf einzelnen Laserdaten gibt ein relativ eindeutiges Ergebnis. Wie man in Tabelle 4.5 gut sehen kann, ist die Reduktion der Scanpunkte sehr effizient. Allein die Reduktion mit einem Punkt-Abstand von 5 cm bewirkt eine Reduktion der Punkte auf ein Sechstel bei gleichbleibender Qualität. Es ist jedoch zu bemerken, dass die Anzahl der Fehlerfälle steigt. Die Fehlerfälle treten jedoch bei den großen Transformationen auf, welche bei Stützung durch Odometriedaten sehr unwahrscheinlich sind. Während die Anzahl der Scanpunkte annähernd logarithmisch zu fallen scheint, steigt die Anzahl der Fehlerfälle linear mit der Filtergröße. Ab einem Punktabstand von mehr als 10 cm scheint die Qualität der Transformationen sprunghaft zu sinken. Die optimale Reduktionsfiltergröße scheint also zwischen 5 cm und 10 cm zu liegen.

<i>pointDist</i>	Punkte	Fehlerfälle	<i>X</i> -Abweichung	<i>Y</i> -Abweichung	θ -Abweichung
0	646	3 / 510	10.1732	5.61099	0.848439
50	109	6 / 510	7.69124	4.88827	0.576586
100	58	9 / 510	8.41346	4.21686	0.603049
150	43	13 / 510	40.9559	43.3175	8.48156

Tabelle 4.5: Testergebnisse bei wechselndem Parameterwert für *reducedPointDist*. Für die Berechnung der verwendeten Scanpunkte ist nur ein beliebiger Beispielscan aus einer RoboCup Arena verwendet worden.

Evaluation auf ganzen Karten Die Evaluation auf ganzen Karten stützt das bisherige Ergebnis. Jedoch scheint sich eine größere Wahl des Reduktionsparameters nicht so negativ auf die Kartequalität auszuwirken wie dies bei den Einzeltests der Fall war. Wahrscheinlich hängt dies damit zusammen, dass die großen Transformationen mit Odometrieunterstützung unwahrscheinlich sind. In Abbildung 4.10 sind die Testergebnisse im Detail aufgeführt.

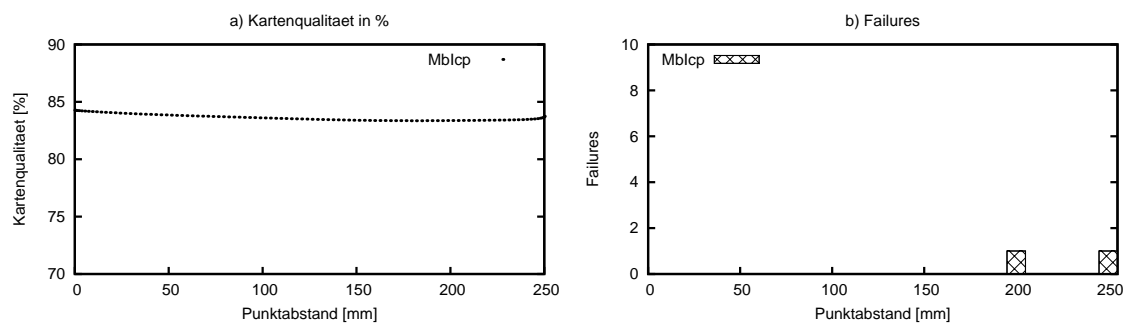


Bild 4.10: Die Kartenqualität und auftretende Fehlerfälle bei wechselnder Reduktionsfiltergröße.

Parameter	Wert
<i>maxRemainingTranslation</i>	1.0
<i>maxRemainingRotation</i>	0.001
<i>maxCorrespondenceDist</i>	100 - 1500
<i>maxScanAngleDist</i>	50.0°
<i>minReducedPointDist</i>	50.0
<i>maxInterpolationDist</i>	150.0
<i>metricFactor</i>	1800
<i>deletedCorrespondences</i>	0% bzw. 20%

Tabelle 4.6: Parameterbelegung zur Evaluation der Ausreißer Eliminierung

4.1.5 Maximale Distanz der Korrespondenzpunkte

Versuchsaufbau

Die Wahl eines maximalen Abstands zwischen zwei möglichen Korrespondenzpartnern führt zu mehr Effizienz bei der Korrespondenzbestimmung und besseren Ergebnissen. Sollte dieser Wert zu gering gewählt werden, werden größere Transformationen wahrscheinlich nicht mehr korrekt berechnet. Daher ist die korrekte Wahl dieses Parameters entscheidend für die Qualität des Ergebnisses.

Versuchsdurchführung

Da dieser Wert abhängig von der realen Roboterbewegung ist, bietet es sich an ihn auf der ganzen Karte zu evaluieren. Verwendet wurde hierzu das Logfile vom RoboCup 2008 in Hannover. Die angewandten Programmparameter sind in Tabelle 4.6 nachzulesen

Ergebnisse

Überraschenderweise sinkt die Kartenqualität rapide durch das Zulassen weit entfernter Korrespondenzen. Die besten Ergebnisse liefern Beschränkungen von ungefähr 10 cm. Dies liegt wahrscheinlich an der Qualität der Odometriedaten und der kurzen Zeitinter-

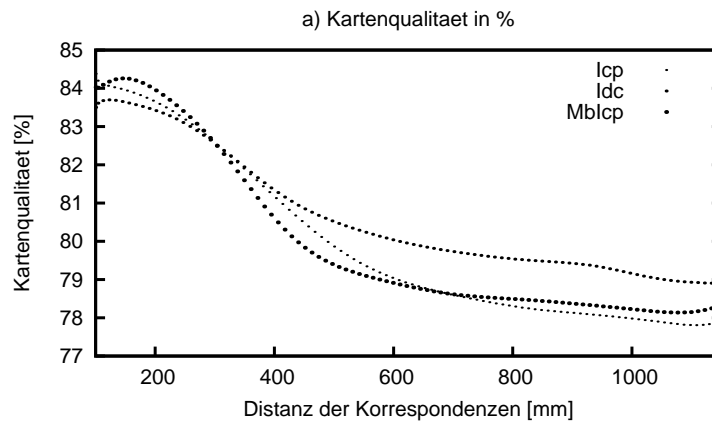


Bild 4.11: Vergleich der Qualität bei Variation der maximalen Korrespondenz-Distanz.

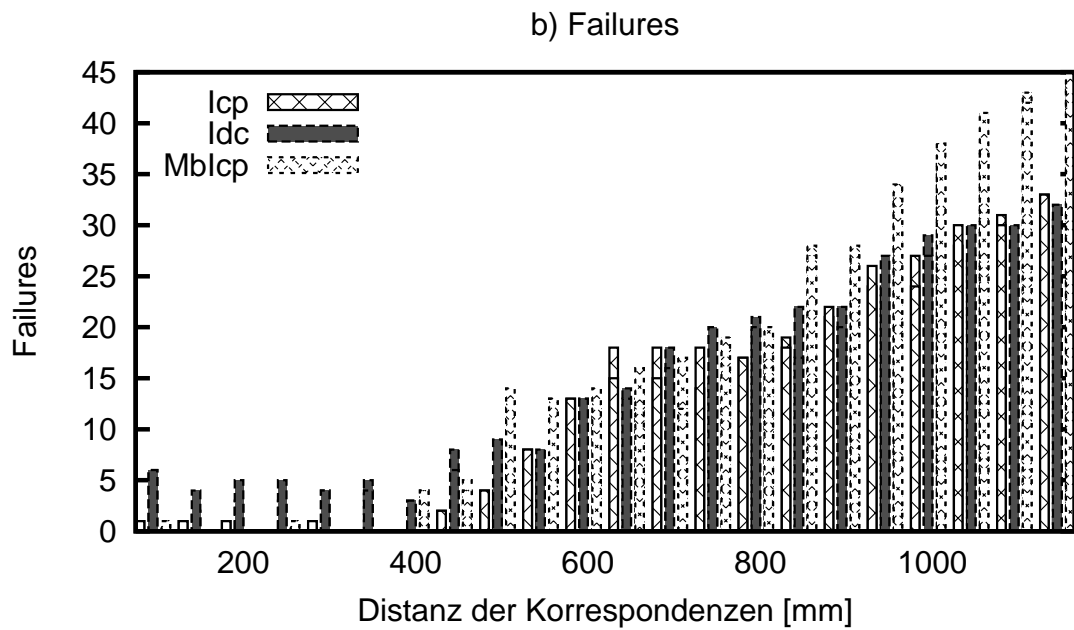


Bild 4.12: Vergleich der Fehlerfälle bei Variation der maximalen Korrespondenz-Distanz.

valle zwischen den Mapping-Schritten. Da das Scanmatching jedoch hauptsächlich dann benötigt wird, wenn die Odometriedaten versagen, sollte dieser Wert größer gewählt werden. Die genauen Testergebnisse sind in den Diagrammen 4.11 und 4.12 einzusehen. Nach diesen Ergebnissen bietet sich ein Parameterwert zwischen 20 cm und 30 cm an.

4.1.6 Maximale Winkeldistanz der Korrespondenzpunkte

Versuchsaufbau

Zur Einschränkung der Korrespondenzsuche und zur Qualitätssicherung wird eine maximale Winkeldistanz festgelegt. Diese Distanz wird für jede Iteration neu berücksichtigt. Daher ist es möglich, dass Iterationen berechnet werden, die nicht innerhalb der festgelegten Winkeldistanz liegen.

Es ist zu erwarten, dass eine zu gering gewählte Winkeldistanz das Verfahren tendentiell eher zu einem kleinen lokalen Minimum führt, wohingegen ein zu weit gewählte Winkeldistanz dieses überspringen aber zu einem entfernten lokalen Minimum übergehen kann.

Auch sollte man beachten, dass der *Idc* den abgesuchten Winkelbereich mit jeder Iteration automatisch verringert. Die anderen Verfahren machen dies nicht.

Versuchsdurchführung

Auch dieser Parameter ist abhängig von der realen Roboterbewegung und wird daher auf der ganzen Karte evaluiert. Hierzu wurde das Logfile des Autonomie-Finales des RoboCups 2008 in Hannover abgespielt und die Karten automatisch bewertet. Tabelle 4.7 zeigt die verwendeten Programmparameter.

Ergebnisse

Die Wahl der maximalen Winkeldistanz zur Korrespondenzbestimmung scheint keinen bedeutsamen Einfluss auf die Qualität des Scanmatchings zu haben. Für *Icp* und *Mblcp* weist die Kartenqualität ab ungefähr 15° kaum Unterschiede auf. Abbildung 4.13 zeigt die

Parameter	Wert
<i>maxRemainingTranslation</i>	1.0
<i>maxRemainingRotation</i>	0.001
<i>maxCorrespondenceDist</i>	100
<i>maxScanAngleDist</i>	10° - 90°
<i>minReducedPointDist</i>	50.0
<i>maxInterpolationDist</i>	150.0
<i>metricFactor</i>	1800
<i>deletedCorrespondences</i>	0% bzw. 20%

Tabelle 4.7: Parameterbelegung zur Evaluation der Ausreißer Eliminierung

automatisch bestimmten Kartenbewertungen im Vergleich. Lediglich der *Idc* scheint eine größere Winkeldistanz zu bevorzugen, wie man anhand der Fehlerquoten in Diagramm 4.14 sehen kann.

Da die Resultate der anderen Scanmatching-Verfahren mit größerer Winkeldistanz auch nicht schlechter werden, empfiehlt es sich den Parameter nicht zu klein zu wählen. Ein Parameter größer 45° bietet jedoch auch keinen Nutzen, da ab Rotationen über 45° in einer rechtwinkligen Umgebung falsche Zuordnungen allgemein sehr wahrscheinlich sind.

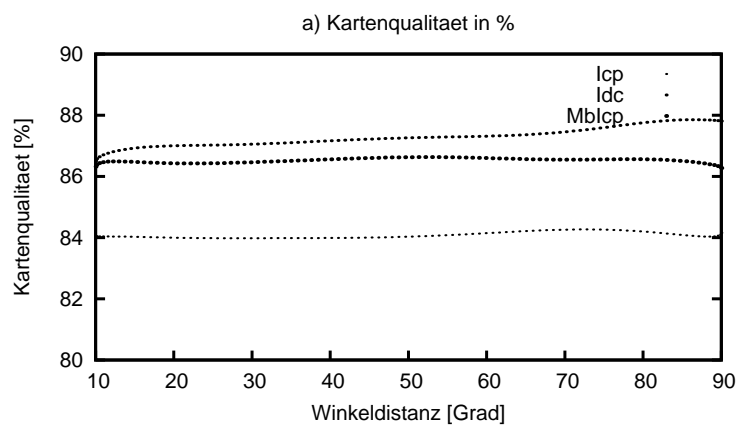


Bild 4.13: Vergleich der Qualität bei Veränderung der maximalen Winkeldistanz.

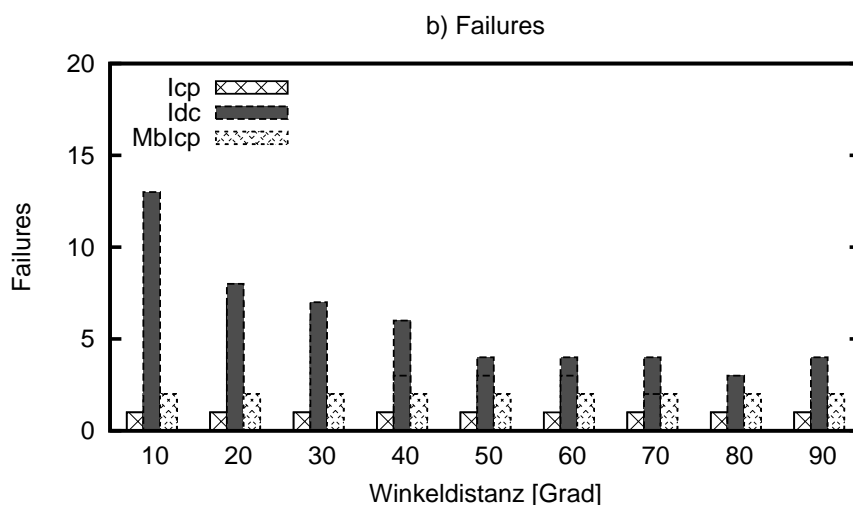


Bild 4.14: Vergleich der Fehlerfälle bei Veränderung der maximalen Winkeldistanz.

4.2 Vergleich der Verfahren

Versuchsaufbau

In diesem Abschnitt werden die unterschiedlichen Verfahren gegeneinander getestet. Ziel hierbei ist es, die Stärken und Schwächen der Algorithmen ausfindig zu machen. Sollten hierbei gravierende Unterschiede identifiziert werden, so ist es möglich je nach Situation einen passenden Scanmatcher zu wählen.

Versuchsdurchführung

Die Evaluation findet sowohl auf einzelnen Daten als auch auf ganzen Karten statt. Die Parameter sind jeweils in Tabelle 4.8 und ?? zu finden.

Ergebnisse

Evaluation auf ausgewählten Laserdaten Die Tests auf einzelnen Laserdaten zeigen, dass die Qualität der aller Scanmatchingverfahren ungefähr gleich gut ist. In den Abbil-

Parameter	Wert
<i>maxRemainingTranslation</i>	1.0
<i>maxRemainingRotation</i>	0.001
<i>maxCorrespondenceDist</i>	800.0
<i>maxScanAngleDist</i>	80.0
<i>minReducedPointDist</i>	50.0
<i>maxInterpolationDist</i>	150.0
<i>metricFactor</i>	1800
<i>deletedCorrespondences (Icp, MbIcp)</i>	0%
<i>deletedCorrespondences (Idc)</i>	20%

Tabelle 4.8: Parameterbelegung zur Evaluation der Verfahren auf ausgewählten Scandaten.

Parameter	Wert
<i>maxRemainingTranslation</i>	1.0
<i>maxRemainingRotation</i>	0.001
<i>maxCorrespondenceDist</i>	200.0
<i>maxScanAngleDist</i>	40.0
<i>minReducedPointDist</i>	50.0
<i>maxInterpolationDist</i>	150.0
<i>metricFactor</i>	1800
<i>deletedCorrespondences (Icp, MbIcp)</i>	0%
<i>deletedCorrespondences (Idc)</i>	20%
<i>maxPointDistance</i>	50.0
<i>minPointsOnLine</i>	10
<i>minLineLength</i>	100.0
<i>maxLineSigma</i>	15.0

Tabelle 4.9: Parameterbelegung zur Evaluation der Verfahren auf ganzen Karten. Die letzten vier Parameter werden zur der Linienextraktion verwendet. Für ihre Bedeutung siehe Kapitel 3.2.4.

dungen 4.15 und 4.16 kann man erkennen, dass die berechneten Transformationen kaum von den gesuchten abweichen. Lediglich der *Icp* weist bei großen Rotationen viele Fehlerfälle auf, in denen keine Transformation bestimmt werden konnte.

Den größten Unterschied sieht man im Laufzeitverhalten. Der *Icp* benötigt im Durchschnitt doppelt so viele Iterationen wie der *Idc*. Der *MbIcp* liegt im Mittelfeld. Die geringe Iterationszahl des *Idcs* verwischt jedoch ein wenig die wahre Laufzeit. Der *Idc* berechnet pro Durchlauf zwei Korrespondenzen und zwei Transformationen. Der Schritt der Korrespondenzbestimmung ist der aufwenigste der Scanmatchingverfahren. Somit terminiert der *MbIcp* mindestens genauso schnell wie der *Idc*.

Evaluation auf ganzen Karten Bei der Evaluation auf ganzen Karten wurden nun auch die Linien-basierten Verfahren mit einbezogen. Betrachtet man die Diagramme in Abbildung 4.17, so erkennt man auf den ersten Blick die Überlegenheit der Punkt-basierten Verfahren. Vor allem das Linien-basierte Verfahren des *Idcs* fällt gewaltig hinter dem Punkt-basierten zurück.

Der Vergleich der Algorithmen untereinander ist schwierig. In Qualität und Terminationsverhalten sind nur geringfügige Unterschiede festzustellen. Das *MbIcp*-Verfahren liefert jedoch die geringste Anzahl an Fehlerfällen, dicht gefolgt vom *Icp*-Algorithmus.

Die Ergebnisse des *MbIcps* sind ein wenig besser als die des *Icps*. Desweiteren hat er auch während der vorhergegangenen Tests den robustesten Eindruck hinterlassen. Vor allem bei der Wahl unterschiedlicher Konvergenzkriterien zeigte er große Überlegenheit gegenüber den anderen Verfahren. Somit liefert die Wahl des *MbIcps* als einzigen Scanmatcher die besten Ergebnisse bezüglich Qualität und Robustheit.

4.3 Integration in Partikelfilter

Versuchsaufbau

Die bisherigen Tests auf Karten wurden mit Unterstützung der Odometriedaten durchgeführt. Um die besten Karten zu erhalten, ist es notwendig auch den Partikelfilter mit

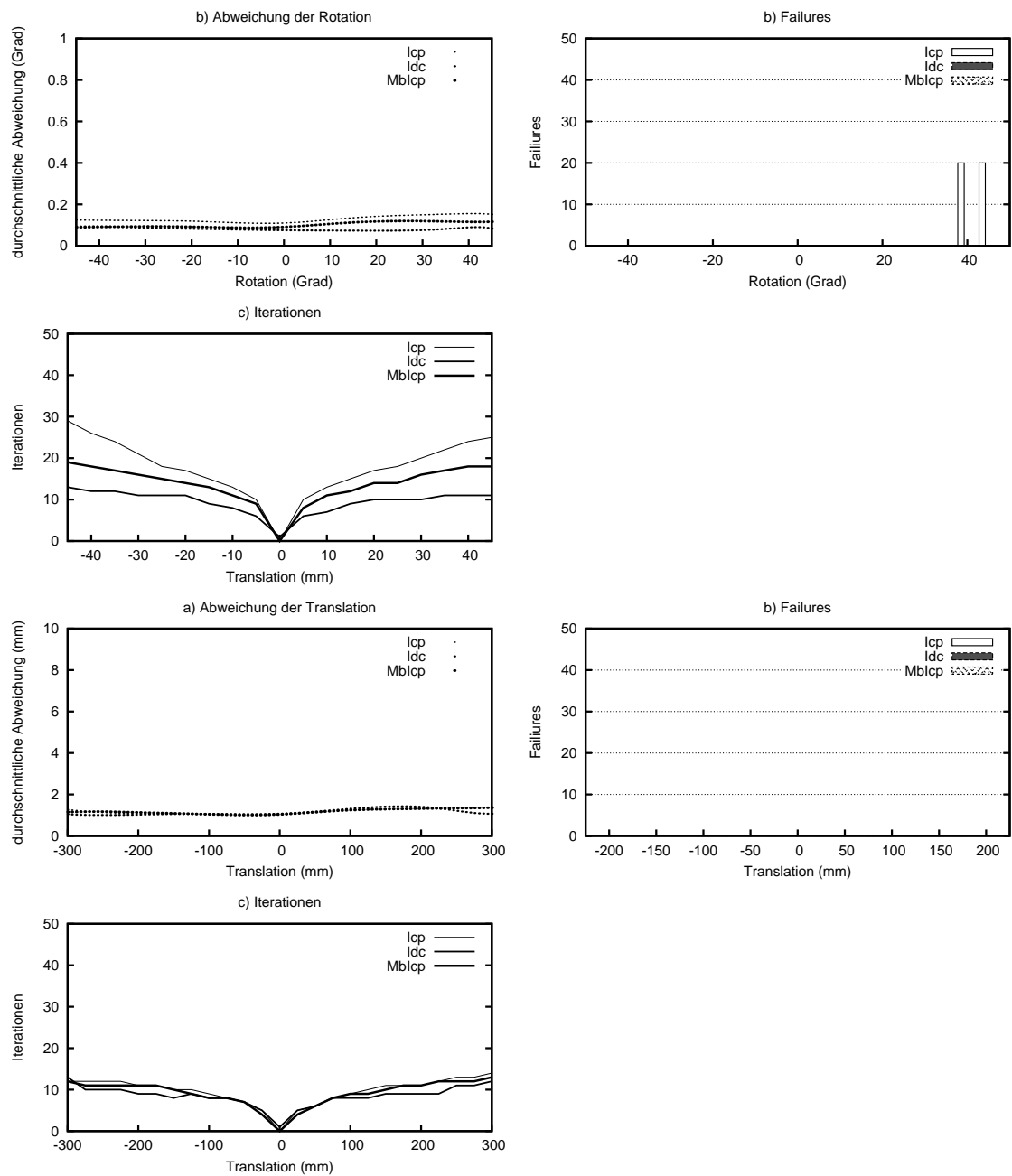


Bild 4.15: Vergleich der Scanmatching-Verfahren bei ausschließlicher Rotation und Translation.

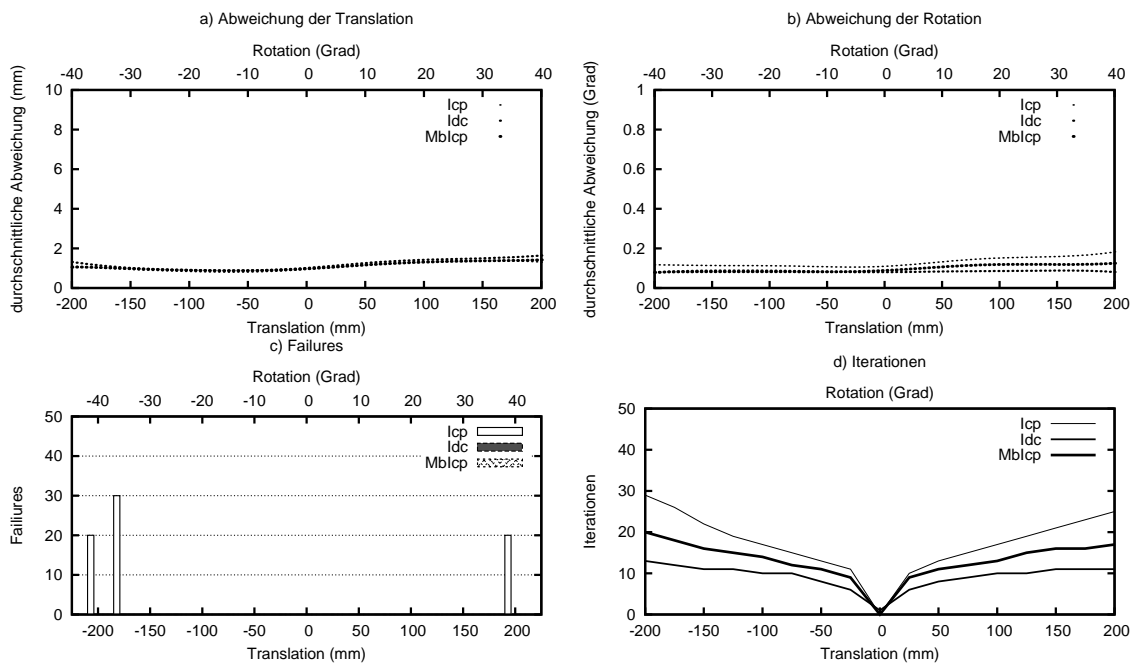


Bild 4.16: Vergleich der Scanmatching-Verfahren bei gleichzeitiger Rotation und Translation

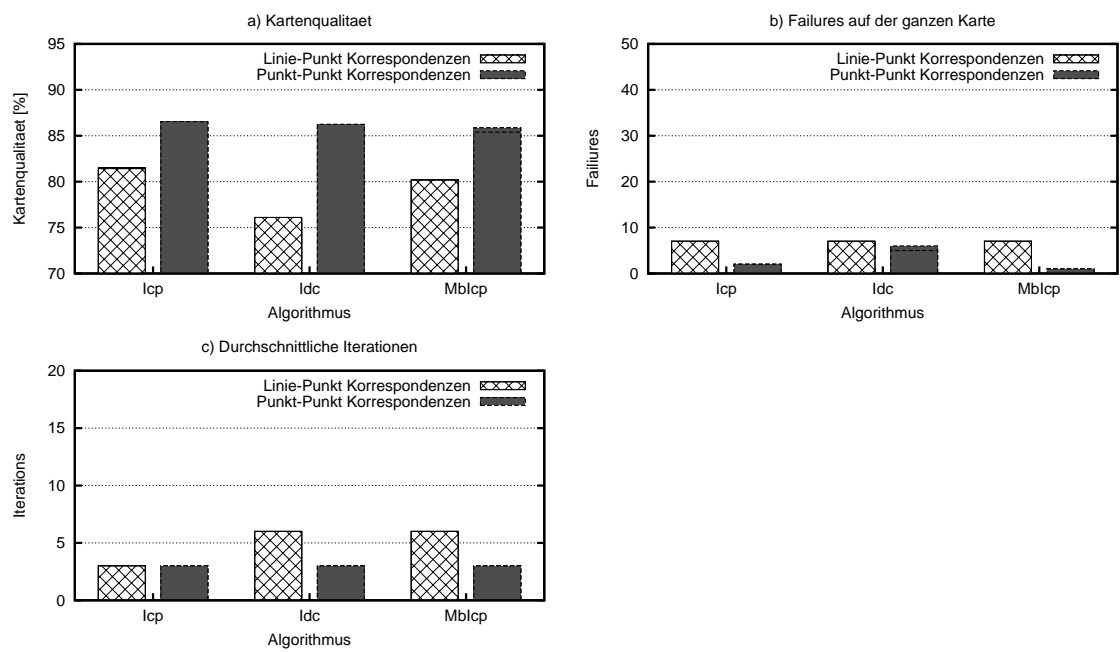


Bild 4.17: Vergleich der Scanmatching Verfahren. Icp, Idc, Mblcp jeweils mit Punkt-Punkt und Punkt-Linie Korrespondenzen.

einzu beziehen. Zeitgleich mit dieser Studienarbeit wurde die Effizienz des Partikelfilters enorm gesteigert. So ist es nun möglich die Anzahl der bisherigen 300 Partikel problemlos auf 3000 zu erhöhen. Dies stellt die Bedeutung des Scanmatchers voraussichtlich weit zurück.

Im Rahmen dieses Tests sollen die Verbesserung evaluiert werden, die durch den Einsatz des Scanmatchers zustande kommt. Hierzu wird die Anzahl der Partikel laufend erhöht, um den Unterschied zwischen dem Partikelfilter mit Scanmatcher und dem Partikelfilter ohne Scanmatcher herauszuarbeiten.

Versuchsdurchführung

Der Test wurde auf ganzen Karten evaluiert. Beginnend mit einem Partikel und somit den reinen Odometriedaten wurde die Anzahl der Partikel pro Testdurchlauf um jeweils 50 erhöht. Die Test wurden für den Partikelfilter jeweils mit und ohne Scanmatcher durchgeführt.

Ergebnisse

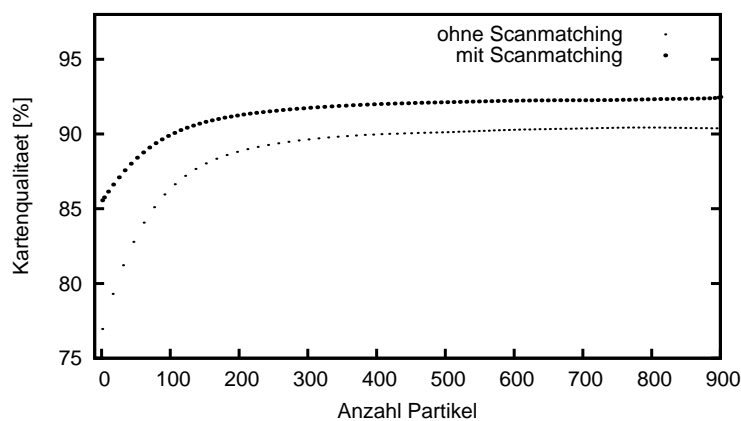
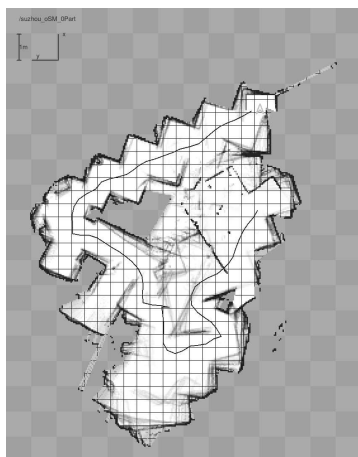


Bild 4.18: Vergleich eines Partikelfilters mit Scanmatchings mit einem ohne bei steigender Partikelzahl.

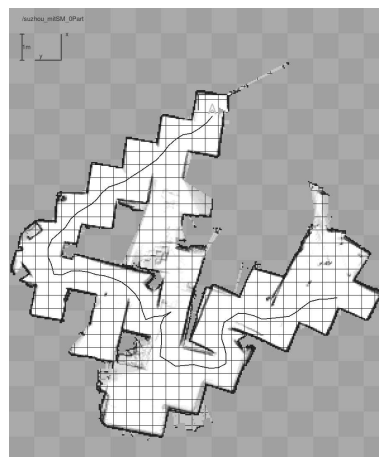
Die Resultate es Scanmatchers sind erfreulich gut. Im Vergleich zu den reinen Odometrie-daten bietet der der Partikelfilter eine sprunghafte Verbesserung von 75% auf 85% Kartenqualität. Mit dem Einsatz des Partikelfilters schrumpft dieser Vorsprung zwar rapide, jedoch sinkt er während der Testreihen niemals unter 2%. In Abbildung 4.18 sind die vollständigen Testreihen zu sehen. Zwei Prozent auf solch hohem Niveau können durchaus einen entscheidenden Unterschied ausmachen.

Zur optischen Beurteilung sind in Abbildung 4.19 einige Karten mit und ohne Einsatz des Scanmatchers gegeben. In linken Spalte befinden sich die Karten, die ohne Verwendung des Scanmatchers erstellt wurden. Die Karten unter Verwendung des Scanmatcher erstellten Karten werden in der rechten Spalte angezeigt.

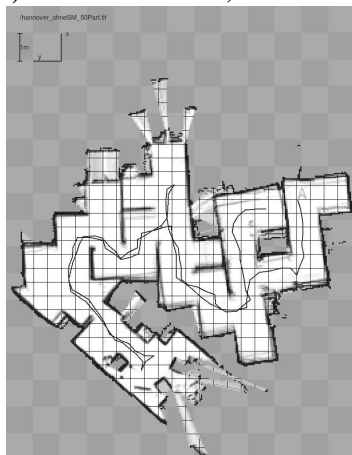
Auf den oberen beiden Karten ist der Unterschied am deutlichsten zu sehen. Da nur einem Partikel verwendet wurde, fand die Erstellung die Karten somit mit reinen Odometriedaten statt. Bei höheren Partikelzahlen sind die Unterschiede meist nur lokal im Detail zu erkennen. Fairerweise muss man anmerken, dass die Kartenerstellung auch von der zufälligen Auswahl der Daten beeinflusst wird und somit bei gleichbleibender Partikelzahl Karten von unterschiedlicher Qualität zustande kommen können. Die Tests in Diagramm 4.18 wurden daher mehrfach durchgeführt und die Ergebnisse gemittelt.



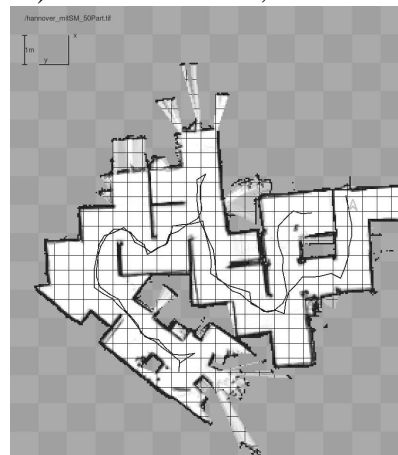
a) o. Scanmatcher, 1 Partikel



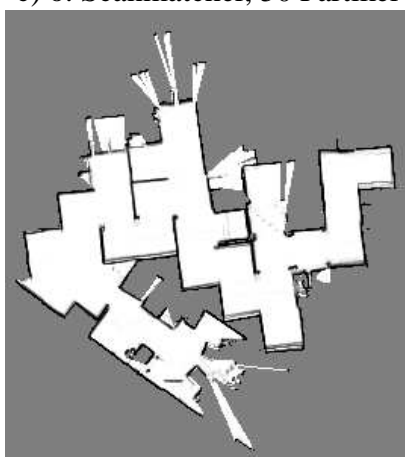
b) m. Scanmatcher, 1 Partikel



c) o. Scanmatcher, 50 Partikel



d) m. Scanmatcher, 50 Partikel



e) o. Scanmatcher, 900 Partikel



f) m. Scanmatcher, 900 Partikel

Bild 4.19: Karten zum optischen Vergleich mit und ohne Scanmatcher bei steigender Partikelzahl.

Kapitel 5

Zusammenfassung

Im Rahmen dieser Studienarbeit wurden die Scanmatching-Verfahren *Icp*, *Idc* und *MbIcp* implementiert und erfolgreich in die Kartenerstellung des mobilen Systems *Robbie* eingebunden. Durch das Hinzufügen einiger Filter und Vorverarbeitungsschritte konnte die Effizienz aller Verfahren merklich verbessert werden.

Evaluationsergebnisse Bei der Evaluation fiel die Qualität der Resultate sehr zufriedenstellend aus. Alle Verfahren sind in der Lage, eine gesuchte Transformation bis auf 1 mm der Translation und $0,1^\circ$ der Rotation genau zu bestimmen. Bei ungünstiger Ausgangsposition ist es jedoch möglich, dass die Scanmatching Verfahren in einem lokalen Minimum der Transformationsbestimmung konvergieren. Dies tritt unterschiedlich oft auf und ist das Hauptmerkmal für die Bestimmung der Robustheit eines Verfahrens.

Das *Icp* Verfahren arbeitet zuverlässig und stabil. Es zeigt lediglich bei großen Rotationen ab 40° Schwächen und tendiert dann gelegentlich zu einem lokalen Minimum. Bei großen Transformationen benötigt der *Icp* jedoch auch eine hohe Anzahl an Iterationen.

Der *Idc* ist das empfindlichste der Verfahren. Es benötigt die meisten einschränkenden Kriterien bei der Bestimmung der Korrespondenzpunkte. Dadurch erreicht es eine sehr schnelle Konvergenz und terminiert im Schnitt dreimal schneller als der *Icp*. Dies bezieht sich jedoch auf die Anzahl der Iterationen - durch die zusätzlichen Berechnungen verliert diese Differenz stark an Gewicht.

Der *MbIcp* erweist sich als das beste der drei Scanmatching-Verfahren. Er hat dem *Icp* gegenüber den Vorteil der schnelleren Konvergenz und der größeren Stabilität bei weiträumigen Transformationen. Den *Idc* schlägt er ebenfalls in beiden Kriterien. Doch den Vergleich gegen den *Icp* gewinnt keineswegs so deutlich wie von [?] behauptet.

Ernüchternd sind die Evaluationsergebnisse der Linien-basierten Verfahren. Sie schneiden in nahezu allen Bereichen schlechter ab als ihre Punkt-basierten Gegenstücke. Vor allem der *Idc* ist aufgrund der Art seiner Korrespondenzbestimmung nicht zum Arbeiten auf Linien geeignet.

Einbindung in das mobile System Robbie Die Scanmatching Verfahren wurden erfolgreich in die Kartenerstellung des mobilen Systems Robbie eingebunden. Sie liefern eine deutlich bessere lokale Korrektheit als die der Odometriedaten. Zur Sicherstellung der globalen Korrektheit ist jedoch die Integration in den Partikelfilter nötig. Durch den Einsatz des Scanmatchings können die Fehlerprobabilitäten des Partikelfilters reduziert werden. Erfreulicherweise ist der Qualitätsgewinn durch das Scanmatching auch noch bei einer großen Anzahl an Partikeln sichtbar.

Ein situationsabhängiges Verfahren wurde als nicht sinnvoll erachtet, da keine signifikanten situationsabhängigen Vorteile der einzelnen Verfahren erkennbar sind. Die Verfahren wurden mittels Strategy-Pattern implementiert und sind somit austauschbar. Das in der Kartenerstellung verwendete Verfahren kann über die Konfigurationseinstellungen ausgewählt werden.

Kapitel 6

Ausblick

In der Zukunft bleibt es zu beobachten, wie sich der Scanmatcher in anderen Umgebungen behauptet. Spannende Einsatzgebiete bieten sich auch außerhalb von RoboCup Rescue. Allein schon in RoboCup@Home ist ein ganz anderer Schwierigkeitsgrad gegeben durch offenere Räume und dynamische Umgebungen.

In der Kartierung von echten Gebäuden kann der Scanmatcher eventuell auch seinen Nutzen zeigen. Da die Fehlerprobabilitäten des Partikelfilters reduziert werden können, werden eventuell typische Kartierungsfehler vermeidbar, wie zum Beispiel die Reduzierung einer Wanddicke zu einem dünnen Strich.

Die Probleme des Scanmatchers sind jedoch auch nicht zu vernachlässigen. Durch Fehler in den Laserdaten wie Verzerrungen durch schrägstehenden Laserscanner, kurzes Mitscannen des Bodens oder das Auftauchen neuer Wände im Scanradius können durchaus Fehler auftreten. Auch die Termination des Verfahrens in einem lokalen Minimum lässt sich niemals ganz verhindern. Man kann lediglich versuchen, durch kurze Zeitabstände zwischen den Mapping-Schritten die Fehlerquote gering zu halten. Außerdem ist es möglich, dass die Parametereinstellungen bei geändertem Fahrverhalten des Roboters angepasst werden müssen.

Dipl. Inf. Johannes Pellenz arbeitet zur Zeit an einem Partikelfilter von Partikelfiltern. Hierbei laufen mehrere Partikelfilter konkurrierend zueinander und sollen in Zukunft auch genetisch erstellt werden. In diesem Rahmen wäre es denkbar, den Einsatz des Scan-

matchers und seine Parameter als wichtige Attribute zur Vererbung zu verwenden. Es ist umgebungs- und situationsabhängig, ob die Kartenerstellung besser mit Scanmatcher oder mit großen Fehlerprobabilitäten des Partikelfilters arbeitet. Durch die konkurrierende Anwendung könnte sich das System optimal an seine Umgebung anpassen.

Verzeichnis der Bilder

2.1	Icp: Korrespondenzzuordnungen bei Rotation und Translation	12
2.2	Vergleich: Zuordnungen Punkt-Punkt und Punkt-Linie	13
2.3	IMRP: Korrespondenzbestimmung nach der gleichen-Entfernungs-Regel .	16
2.4	Idc: Korrespondenzzuordnungen bei Rotation und Translation	19
2.5	MbIcp: Iso-Kurven der neuen Metrik	22
2.6	MbIcp: Korrespondenzzuordnungen bei Rotation und Translation	27
3.1	Klassendiagramm: Scan-Matcher Paket	32
3.2	StateChart: Ablauf Scanmatching	33
3.3	Aktivitätsdiagramm: Vorbereitung der Scan-Punkte	35
3.4	Aktivitätsdiagramm: Vorbereitung der Scan-Linien	35
3.5	Filter: Löschen ungültiger Punkte	37
3.6	Filter: Extraktion von Linien	39
3.7	Filter: Projektionsfilter	41
3.8	Filter: Sichtbarkeitsfilter auf Punkten	43
3.9	Filter: Sichtbarkeitsfilter auf Linien	43
3.10	Filter: Reduzierung der Scan-Punkte	45
3.11	Vergleich: verwendete Koordinatensysteme	49

3.12	Statechart: Kartenerstellung mittels <i>SLAM</i> Verfahren.	52
3.13	Statechart: Selbstlokalisierung mittels Partikelfilter	53
3.14	Aktivitätsdiagramm: Einbindung des Scanmatchings im Partikelfilter	55
3.15	Klassendiagramm: Scan-Matcher und Partikelfilter	56
4.1	Testdaten für die Evaluation auf ausgewählten Datensätzen	62
4.2	Testdaten zur Evaluation auf ganzen Karten	63
4.3	Test: Konvergenzkriterien auf einzelnen Daten, Qualität und Fehlerfälle	66
4.4	Test: Konvergenzkriterien auf Karte, Qualität, Fehlerfälle, Iterationen	67
4.5	Test: metrischer Faktor, Fehlerfälle	69
4.6	Test: metrischer Faktor, Qualität	69
4.7	Test: Eliminierung falscher Punkt-Korrespondenzen, Fehlerfälle	72
4.8	Test Eliminierung falscher Punkt-Korrespondenzen: Qualität	73
4.9	Test Eliminierung falscher Punkt-Korrespondenzen: Iterationen	74
4.10	Test: Reduktionsfilter	76
4.11	Test: maximale Korrespondenz Distanz, Qualität	78
4.12	Test: maximale Korrespondenz Distanz, Fehlerfälle	78
4.13	Test: maximale Winkeldistanz der Korrespondenzen, Qualität	80
4.14	Test: maximale Winkeldistanz der Korrespondenzen, Fehlerfälle	81
4.15	Vergleich der Scanmatching Verfahren mittels ausgewählter Laserdaten 1	84
4.16	Vergleich: Scanmatching Verfahren mittels ausgewählter Laserdaten 2	85
4.17	Vergleich: Scanmatching Verfahren mittels ganzer Karten	86
4.18	Vergleich: Partikelfilter mit steigender Partikelzahl	87
4.19	Vergleich: Karten mit und ohne Scanmatcher	89

Literaturverzeichnis

- [BM92] BESL, Paul ; MCKAY, Neil: A Method for Registration of 3-D Shapes. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence* 14 (1992), Nr. 2, 239-256. <file:///lab/as/Docs/Articles/Besl1992AMF.pdf>
- [GS96] GUTMANN, Jens-Steffen ; SCHLEGEL, Christian: AMOS: Comparison of Scan Matching Approaches for Self-Localization in Indoor Environments. In: *1. Euro-micro Workshop on Advanced Mobile Robots*. Kaiserslautern : IEEE Computr Society Press, 1996, S. 61–67
- [Gut00] GUTMANN, Jens-Steffen: *Robuste Navigation autonomer mobiler Systeme*, Albert-Ludwigs-Universität Freiburg, Diss., 2000
- [IB98] ISARD, M. ; BLAKE, A.: CONDENSATION - conditional density propagation for visual tracking. In: *International Journal of Computer Vision* 29 (1998), Nr. 1, 5-28. <ftp://ftp.robots.ox.ac.uk/pub/ox.papers/VisualDynamics/ijcv98.ps.gz>
- [LM94] LU, Feng ; MILIOS, Evangelos: Robot pose estimation in unknown environments by matching 2d range scans. In: *IEEE Computer Vision and Pattern Recognition* (1994)
- [LM97] LU, F. ; MILIOS, E.: Globally Consistent Range Scan Alignment for Environment Mapping. In: *Autonomous Robots* (1997), Nr. 4, S. 333–349

- [MLM05] MINGUEZ, Javier ; LAMIRAUX, Florent ; MONTESANO, Luis: Metric-Based Scan Matching Algorithms for Mobile Robot Displacement Estimation. In: *ICRA*, IEEE, 2005, 3557-3563
- [NÖ6] NÜCHTER, Andreas: *Semantische dreidimensionale Karten für autonome mobile Roboter*, Rheinische Friedrich-Wilhelms-Universität Bonn, Diss., 2006. <https://www.uni-koblenz.de/~agas/Pool/Nuechter2006SDK.pdf>