

Kapitel 1

Umsetzung

1.1 Ansätze

1.1.1 Iterative Closest Point

Der Iterative Closest Point Algorithmus (ICP) ist ein sehr bekannter und verbreiteter Algorithmus für Rigid Registration. Er funktioniert wie folgt:

Zuerst werden die Transformationsparameter so initialisiert, dass die beiden Punktwolken keine zu großen Unterschiede aufweisen. Danach werden die Transformationsparameter (Rotation, Translation und Skalierung) auf eine Punktwolke angewandt, sodass nun für jeden Punkt aus der transformierten Punktwolke der dichteste Punkt aus der fixen Punktwolke bestimmt werden kann. Wenn die Zuordnung bestimmt wurde kann nun über die Summe der Abstandsquadrate zwischen den Punkten ein Fehler zwischen den Punktwolken berechnet werden. Ist dieser Fehler klein genug ist die Registrierung abgeschlossen, ist er zu groß werden die Transformationsparameter anhand der gefundenen Zuordnung neu bestimmt und der Algorithmus wird iterativ wiederholt bis er konvergiert.

ICP existiert schon recht lange, deswegen gibt es eine Vielzahl von Erweiterungen, wie z.B. EM-ICP, welches ICP durch den EM-Algorithmus erweitert. So existieren auch Varianten die ICP beschleunigen usw.

Der ICP Algorithmus ist auf Grund seiner einfachen Funktionsweise sehr Performant und leicht zu verstehen. Außerdem lässt er sich wegen seines modularen Aufbaus sehr gut erweitern und verändern, was ihn gut anpassbar für viele Situationen macht. Er hat jedoch auch Nachteile in seiner nativen Form. Die Punktwolken müssen initial bereits annähernd zueinander passen, da ICP nicht die richtige Lösung bestimmen kann. Außerdem sorgen fehlende Punkte in einer der Punktwolken für Probleme, da dies den berechneten Fehler verfälscht.

Der ICP Algorithmus wurde aus folgenden Gründen nicht in dieser Arbeit verwendet:

1. Die Rotation zwischen den beiden Punktwolken kann beliebig sein und ist vorher nicht bestimmbar

2. Es können mehrere Punkte in den Punktwolken fehlen, da diese recht viele Punkte enthalten können
3. Die Zuordnung von Punkten ist eindeutig (0 oder 1), was das erkennen von Fehlern oder das bestimmen einer Vertrauenswürdigkeit sehr schwer macht

1.1.2 Graph-Matching

-Allgemeine Beschreibung

- finden von Subgraphen
- Funktionsweise kurz beschreiben
- Stärken: schnell und genau
- Schwächen: outlier können nicht gefunden werden => falsches Ergebnis
- Grafik? (wenn Platz ist ja)

1.1.3 Coherent Point Drift

-Einleitung passend zu Grundlagen

- Begründung warum CPD verwendet wurde:
- outlier erkennen
- noise Data
- soft matching(0.0 - 0.5 - 1.0)? => Falsche matchings können besser erkannt werden
- Performanz Vergleich? (Wieder finden im Paper im Vergleich zu ICP)

1.2 Implementierung

1.2.1 Programmablauf

Das entstandene Softwaremodul umfasst verschiedene Abläufe, welche zum besseren Verständnis im folgenden erläutert werden.

In Abbildung 3.1??? ist der Ablauf der Software anhand eines Aktivitätsdiagramms dargestellt, dieser wird im folgenden genauer erläutert.

Abbildung 1.1: Aktivitätsdiagramm des Softwaremoduls

Zuerst werden die benötigten Daten von der Brücke als NMEA-Nachrichten eingelesen und in Objekte übersetzt. Dies geschieht durch einen NMEA-Parser von Raytheon Anschütz, welcher in das Modul eingebunden ist.

Im folgenden wird ein Algorithmus ausgewählt, welcher passend zu den verfügbaren Daten den weiteren Ablauf bestimmt. Die hierfür geprüften Daten sind:

1. AIS Daten umliegender Schiffe und ATONs

Tabelle 1.1: Genauigkeitssteigerung durch Korrektur des Zeitverzugs

2. Die Geschwindigkeit des eigenen Schiffes
3. Empfangene AIS-Daten von mindestens drei ATONs
Anhand dieser Daten wird ein Algorithmus ausgewählt, welcher in der jeweiligen Situation das beste Ergebnis erzielt.

Der vorher bestimmte Algorithmus gibt nun an welche der drei Filter auf die Daten angewendet werden müssen, um die für den Algorithmus relevanten Daten zu extrahieren. Diese drei sind: das filtern nach den neuesten AIS und ARPA Daten der Schiffe, das filtern nach den ATONs im AIS und das filtern nach ARPA Daten, welche keine Geschwindigkeit aufweisen.

Zusätzlich zu diesen drei Filtern werden die AIS-Daten mithilfe von Seekarten Informationen abgeglichen. Falls ATONs nicht vom AIS erfasst wurden, werden sie nun ergänzt.

Nachdem anwenden der Filter wird der Zeitverzug der AIS-Daten korrigiert. Dieser Zeitverzug existiert, da AIS-Nachrichten in einer Frequenz von 3s bis 2min gesendet werden können. Die Radar Antenne, welche für die ARPA-Daten zuständig ist benötigt jedoch für eine volle Umdrehung bzw. für das erfassen aller umliegenden Schiffe nur ca. 2s bis 4s. Dieser zeitliche Versatz zwischen den beiden Datensätzen hat einen großen Einfluss auf die Genauigkeit der Zuordnung des Algorithmus.

Im folgenden startet der spezifische Algorithmus, die einzelnen Komponenten des Algorithmus werden im Laufe der Arbeit genauer behandelt, so wird *AIS zum Koordinaten-Ursprung verschieben* in Unterabschnitt 4.2.2 behandelt und wie das initiale Heading den Algorithmus beeinflusst wird in 4.23 erklärt. Das Ergebnis des Algorithmus ist die Zuordnung zwischen den Punkten und der Rotationsunterschied, welcher benötigt wurde um die Punktmengen ineinander zu überführen. Die Zuordnung wird für den in 4.3 erklärten Position-Fix Algorithmus benötigt und die Rotation, welche benötigt wird um beide Punktwolken einander zuzuordnen entspricht dem Heading des Eigenschiffes. Das Heading auf diese Weise zu bestimmen ist möglich, da die ARPA-Daten von der relativen Ausrichtung des Schiffes abhängen, die AIS-Daten jedoch nach Norden ausgerichtet sind. Somit entspricht eine Drehung der ARPA-Daten, bis sie mit den AIS-Daten übereinstimmen, dem Winkel zwischen der Ausrichtung des Schiffes und der Himmelsrichtung Norden. Dies ist der Winkel, in welchem das Heading eines Schiffes angegeben wird.

Mit den Ergebnissen des Coherent Point Drift Algorithmus kann nun der Position-Fix die Eigenposition bestimmen (4.3) und diese als Positionsangabe über eine NMEA-Nachricht an die Brücke senden. Zusätzlich wird noch eine Visualisierung des Ergebnisses über die von Herrn André Becker entwickelte Netzwerkschnittstelle an die ECDIS gesendet.

1.2.2 Ungenauigkeiten durch den Erdellipsoid

Der Coherent Point Drift Algorithmus basiert auf Koordinaten in kartesischer Form. In dieser Arbeit wurde er, auf geodätische Koordinaten angewendet, wodurch einige Genauigkeitsprobleme entstanden, die im weiteren erläutert werden und eine Lösung für diese präsentiert wird.

Das erste Problem beruht auf der Translation, welche der Algorithmus anwendet um die Distanz zwischen zwei Punktwolken zu minimieren.

Die Translation basiert auf dem Unterschied der Punktwolken in x und y Achse. Dies ist zwar korrekt in einem kartesischen Koordinatensystem, wo der Abstand zwischen zwei Punkten über die Euklidische Norm bestimmt werden kann, jedoch nicht wenn mit geodätische Koordinaten gearbeitet wird. Der Grund hierfür ist, dass der Abstand zwischen den Längengraden zwischen Nord- bzw. Südpol und dem Äquator sich je nach Position ändern. Als Folge hiervon ist es zwar möglich die Punktwolken zueinander zu bewegen, jedoch ist die Translation, welche der Algorithmus bestimmt, mit einem Fehler versehen. Diese Abweichungen die durch dieses Problem entstanden sind verhalten sich proportional zum Abstand der beiden Punktwolken (siehe Abbildung/Tabelle ???).

Das zweite Problem entstand durch die Rotation, welche Teil des Algorithmus ist. Bei einer Rotation werden alle Punkte einer Punktwolke um den Ursprung des Koordinatensystems gedreht und erhalten somit eine neue Position. Durch das Rotieren einer kartesischen Punktwolke um den Ursprung ändern sich die Positionen der Punkte, jedoch nicht die Abstände zwischen den Punkten. Wenn man geodätische Koordinaten rotiert ändern sich ebenso die Positionen der Punkte, jedoch ändert sich hier auch der Abstand zwischen den Punkten auf Grund der Unterschiedlichen Abstände zwischen Längengraden. Die Abweichungen die durch dieses Problem entstanden sind verhalten sich proportional zu der Verdrehung zwischen den Punktwolken und proportional zum Abstand zwischen Punktwolke und Ursprung.

Die Lösung für das erste Problem (Translation), ist dass minimieren der initialen Abstände zwischen den Punktwolken. Die Translation die nun benötigt wird für das überführen der einen Punktwolke in die andere ist so gering, dass der Fehler der durch die Erdkrümmung entsteht vernachlässigbar ist.

Die Lösung für das zweite Problem (Rotation) ist, dass verschieben des Schwerpunktes der zu rotierenden Punktwolke in den Ursprung des Koordinatensystems. Eine Rotation dieser Punktwolke bewirkt nun dass der Schwerpunkt seine Position nicht verändert, wodurch die Punkte der Punktwolke eine minimale Positionsänderung durch eine Rotation erfahren.

Somit ist die Lösung für diese Probleme die Translation beider Punktwolken auf den Ursprung des Koordinatensystems, um so die Ungenauigkeiten so gering wie möglich zu halten. Dies hat auch den Effekt das Fehler, welche durch das verwenden der euklidischen Norm auf dem Erdellipsoid entstehen, geringer ausfallen, weil Längen- und Breitengrad im Ursprung des Koordinatensystems fast gleich lang sind.

Für die ARPA - Punktwolke ist dies trivial, da aufgrund der Radar Daten diese Punkte in Polarkoordinaten angegeben sind, was es leicht macht diese zu auf

einen bestimmten Punkt zu verschieben.

Die AIS - Punktwolke ist jedoch an ihre absolute Position gebunden, deswegen ist eine andere Methodik notwendig um die Originale Punktwolke an den Ursprung verschieben zu können. Hierfür wird der Schwerpunkt der AIS - Punktwolke bestimmt, da diese aus nicht gewichteten Punkten besteht ist dieser äquivalent zum arithmetischen Mittel der Punktwolke. Nun wird die Distanz und der Winkel zwischen jedem Punkt der Punktwolke und dem arithmetischem Mittel bestimmt und gespeichert. Für die Berechnung der Distanz und der Winkel zwischen den Punkten wird die Vincenty Formel verwendet, damit keine Fehler durch die Erdkrümmung entstehen. Nun liegt die AIS - Punktwolke in Polarkoordinaten vor. Das macht es möglich sie ebenso wie die ARPA - Punktwolke an den Ursprung zu platzieren.

Nun sind optimale Bedingungen geschaffen um eine Zuordnung zwischen den Punktwolken durch den Coherent Point Drift Algorithmus zu bestimmen.

1.2.3 Rotationsschwäche von Coherent Point Drift

Das erkennen von Rotation ist eine essentielle Eigenschaft, welche der Algorithmus benötigt um die Punktwolken einander zuzuordnen und somit auch das Heading des Eigenschiffes zu bestimmen. Die Rotation kann jeden beliebigen gültigen Wert entsprechen, da die Drehung der ARPA Punktwolke abhängig vom Heading ist und die Drehung der AIS Punktwolke immer nach Norden ausgerichtet ist. Somit ist jeder Rotationswinkel eine mögliche Lösung und das Problem kann nicht vorher behandelt werden.

Bei Experimenten mit dem Coherent Point Drift ist jedoch aufgefallen, dass dieser Rotation nur bis ca. 60° erkennen kann. Bei Punktwolken, welche mehr als 60° Rotationsunterschied besitzen, findet der Algorithmus nur ein lokales Minimum und gibt dies als Lösung zurück. Die Ergebnisse des Experiments sind in Abbildung ??? visualisiert. Diese Schwäche im erkennen der Rotation vom Coherent Point Drift Algorithmus wurde bereits in anderen Arbeiten festgestellt. [HIER PAPER VERLINKEN](#).

Für dieses Problem wurde eine Lösung entwickelt, mit welcher die korrekte Rotation, sowie die korrekte Zuordnung bestimmt werden können.

Hierfür wird der Algorithmus mehrfach ausgeführt, wobei nach jeder Ausführung die ARPA-Punktwolke um 45° rotiert wird und die Ergebnisse gespeichert werden. Die Rotation der ARPA-Punktwolke ist sehr simple, da sie bereits in Polarkoordinaten vorliegt. Nachdem der Algorithmus 8 mal ausgeführt wurde ($\frac{360}{45} = 8$) werden die Ergebnisse verglichen. Hierbei wird das Ergebnis mit dem geringsten Fehler zwischen den Punkten ausgewählt, da dieses das globale Optimum indiziert. Bei Tests mit diesem Ansatz wurde unter verschiedenen Bedingungen immer der korrekte Rotationswinkel und die richtige Zuordnung gefunden.

Diese Lösung hat natürlich einen großen Nachteil, welcher die Performanz des Algorithmus ist, da dieser nun statt einmal pro Durchlauf acht mal aufgerufen wird. Es ist jedoch nur notwendig beim initialen Ausführen des Softwaremoduls den Algorithmus acht mal zu starten, da nach dem ersten Durchlauf des Soft-

waremoduls ein Heading an die Brücke gesendet wurde, welches beim nächsten Aufrufen des Algorithmus wieder mit gesendet werden kann. Mit einem Heading, dass nicht mehr als 50° abweicht kann der Algorithmus ohne Probleme arbeiten und muss nun die ARPA-Punktwolke nur um den Wert des negativen Headings drehen.

1.3 Position-Fix

1.3.1 Bestimmung der Eigenposition

Mit dem Ergebnis des Coherent Point Drift Algorithmus können ARPA - Punkte auf ihre zugehörigen AIS - Punkte zugeordnet werden. Dadurch ist nun bekannt wo dieser Punkt absolut auf der Karte liegt (AIS-Daten) und wie er relativ zu der Schiffseigenposition liegt (ARPA-Daten). Durch diese beiden Daten ist es möglich die Schiffseigenposition für jedes Paar zugeordneter Punkte zu berechnen.

Damit diese Rückrechnung auf die Eigenposition möglich ist, muss zuerst das relative Bearing aus den ARPA - Daten in *True* Bearing umgerechnet werden. Als *True* Bearing wird das Bearing bezeichnet, welches relativ zur Nord-Richtung angegeben wird. Diese Umrechnung ist möglich, da der CPD-Algorithmus auch das Heading des Schiffes bestimmt. Das Heading des Schiffes ist relativ zur Nord-Richtung und bietet somit die Möglichkeit über folgende Formel eine relative Bearing Angabe in *True* Bearing umzurechnen:

b_{true} : *True* Bearing
 b_{rel} : Relatives Bearing
 h : Heading des Eigenschiffes

$$b_{true} = (b_{rel} - h) \mod 360$$

Nun kann von jedem Datenpaar eine individuelle Eigenposition bestimmt werden. Dies basiert auf der Lösung der 1. geodätischen Hauptaufgabe, diese ist das berechnen eines Punktes mit einem Startpunkt, einer Richtung und einer Distanz. Der Startpunkt sind die AIS Koordinaten, die Richtung ist das invertierte *true* Bearing und die Distanz kann aus den ARPA - Daten entnommen werden. Die Lösung der 1. geodätischen Hauptaufgabe lässt sich mithilfe der Vincenty Formel bestimmen.

Nachdem dies für jedes Datenpaar wiederholt wurde, existiert eine Punktwolke, welche die individuellen Eigenpositionen enthält. Zur Bestimmung der Eigenposition wird das arithmetische Mittel über diese Punktwolke gebildet. Falls die einzelnen Punkte vorher eine Gewichtung besitzen, zur Indizierung einer höheren Vertrauenswürdigkeit, kann hier auch das gewichtete arithmetische Mittel angewendet werden.

1.3.2 Bestimmung der Sicherheit der Eigenposition und Eliminierung von Ausreißern

Die Bestimmung der Eigenposition erfolgt über das arithmetische Mittel, dieses ist jedoch anfällig für Ausreißer in der Punktwolke. Zusätzlich ist es notwendig die bestimmte Eigenposition mit einer Sicherheit bzw. eines Sicherheitsbereichs angeben zu können.

Durch die Anfälligkeit des arithmetischen Mittels auf Ausreißer, ist es notwendig diese zu bestimmen und in gravierenden Fällen zu eliminieren. Dies kann jedoch nur angewendet werden, wenn genügend Punkte in der Punktwolke existieren, da sonst Ausreißer nicht eindeutig klassifiziert werden können.

Zur Erkennung der Ausreißer wird die Mahalanobis Distanz verwendet. Die Mahalanobis Distanz ist ein Abstandsmaß, welche angibt wie weit ein Punkt von der Verteilung einer Punktwolke abweicht. Das Ergebnis der Mahalanobis Distanz wird in Standardnormalabweichungen angegeben. Die Mahalanobis Distanz ist wie folgt definiert:

$\vec{X}_i = \begin{pmatrix} lat_i \\ lon_i \end{pmatrix}$: eine Zufallsvariable, welche die zurückgerechnete Position eines Datenpaars beschreibt. (*lat* ist der Breitengrad und *lon* der Längengrad des Punktes)

$\vec{\mu} = \begin{pmatrix} \mu_{lat} \\ \mu_{lon} \end{pmatrix}$: Das arithmetische Mittel der Punktwolke

Σ : Die Kovarianzmatrix der Punktwolke für welche gilt $\det(\Sigma) \neq 0$
Mahalanobis-Distanz:

$$d(\vec{X}, \vec{\mu}) = \sqrt{(\vec{X} - \vec{\mu})^T \cdot \Sigma^{-1} \cdot (\vec{X} - \vec{\mu})}$$

zur Berechnung der Mahalanobis Distanz wird nun das arithmetische Mittel und die Kovarianzmatrix bestimmt, jedoch ohne einbeziehen des Punktes zu welchem die Mahalanobis Distanz bestimmt werden soll, da sonst das Ergebnis verfälscht wird. Nachdem dies für jeden individuelle Eigenposition berechnet wurde ist es möglich Punkte zu entfernen, welche außerhalb einer bestimmten Prozentzahl der Verteilung liegen. Für diese Prozentzahl wurde 99,865% gewählt, das entspricht einem Z-Wert der Standardnormalverteilung von 3. Somit werden Punkte entfernt die Außerhalb von 99.865% der Verteilung liegen.

Nachdem die Ausreißer entfernt wurden ist es wichtig die Genauigkeit des Ergebnisses angeben zu können. Hierfür wurde sich entschieden einen Sicherheitsbereich zu bestimmen, durch welchen eine prozentuale Sicherheit gegeben werden kann, wie wahrscheinlich es ist sich in diesem Bereich zu befinden. Dieser Sicherheitsbereich wird mithilfe der Mahalanobis Distanz bestimmt. Man setzt die Mahalanobis Distanz auf einen festen Wert, wie z.B. 2,33 was ca. 99% Sicherheit entspricht. Nun kann durch einsetzen des arithmetischen Mittels und der Kovarianzmatrix eine Menge an Punkten bestimmt werden, welche diese Mahalanobis Distanz haben. Die Form der Punkte entspricht einer Ellipse, welche der Sicherheitsbereich ist. Eine Visualisierung dieses Ablaufs ist in Abbildung????

zu erkennen.

1.3.3 Erstellung des Ergebnisses

Nach dem Position-Fix Algorithmus wurden alle benötigten Daten berechnet um die Eigenposition, das Heading und eine geeignete Visualisierung darzustellen. Die Eigenposition und das Heading werden als NMEA-Nachricht an die Brücke gesendet, da ihre Daten so intern wie die Daten eines Sensors behandelt werden und dadurch eine Integritätsprüfung im Vergleich zu den anderen Sensoren möglich ist. Die Visualisierung und zusätzliche Daten, wie der Sicherheitsbereich, werden über die Netzwerkschnittstelle von Herrn André Becker an die ECDIS übermittelt. Die Visualisierung ist eine SVG-Grafik, welche beispielhaft in Abbildung ??? abgebildet ist.

Kurze Erklärung wie das Ergebniss erstellt wird (NMEA Telegram und Grafik) um von der Netzwerkschnittstelle gesendet werden zu können.