

Erstellen einer elektronischen Flußkarte aus Radarbildsequenzen

Martin Sielaff

16. Oktober 1996

Inhaltsverzeichnis

1	Einleitung	1
1.1	Das Schiffsprojekt	1
1.2	Aufgabenstellung	2
1.3	Übersicht	3
2	Methoden der Kartenerstellung	4
2.1	Überblick	4
2.2	Bewertung	6
3	Radarbildvorverarbeitung	7
3.1	Einführung in die Morphologie	7
3.1.1	Dilation und Erosion	7
3.1.2	Opening und Closing	9
3.2	Das Radarbild	11
3.3	Der Filter <i>morph</i>	11
3.4	Anpassung	12
4	Analyse eines Radarbildes	15
4.1	Auswahl relevanter Objekte	16
4.2	Approximation durch Linien	17
4.3	Bewertung der Ergebnisse	24

5	Vereinigung der Analyseergebnisse	27
5.1	Darstellung von Radarbild und Karte	27
5.1.1	Datenstrukturen	27
5.1.2	Interne Darstellung	29
5.2	Kalman-Filter	31
5.3	Aktualisierung der Karte	35
5.3.1	Radarbild-Polygonzug mit der Karte kombinieren	36
5.3.2	Überarbeiten der Karte	48
5.4	Beispiel	49
6	Bewertung und Ausblick	51
A	Befehle für den Kommandointerpreter	53
A.1	Das Kommando <code>chartcreate</code>	53
A.2	Das Kommando <code>chartdraw</code>	53
A.3	Das Kommando <code>writepixel</code>	54
A.4	Das Kommando <code>chartobj</code>	54
A.5	Das Kommando <code>approx</code>	54
A.6	Das Kommando <code>chartkalman</code>	55
A.7	Das Kommando <code>chartmatch</code>	55
A.8	Das Kommando <code>chartdel</code>	55
	Literaturverzeichnis	56

Abbildungsverzeichnis

1.1	Elektronische Flußkarte	2
2.1	Arten der Umweltrepräsentation	5
3.1	Morphologischer Filter	10
3.2	Datenstruktur eines Radarbildes	11
3.3	Parameter für den <i>morph</i> -Aufruf	12
3.4	Strukturelement des Filters <i>morph</i>	13
3.5	Filterung von Radarbildern	14
4.1	Objektmerkmale	16
4.2	Objektauswahl, Beispiel Erstechoanteil	17
4.3	Konturbestimmung 1	18
4.4	Konturbestimmung 2	18
4.5	Unterteilung einer Kontur	19
4.6	Auswahl einer Repräsentation anhand der <i>significance</i> -Werte	20
4.7	Das „ <i>significance</i> -Kriterium“	21
4.8	Konturpunkte und approximierende Linie	22
4.9	Approximation der Uferlinien in einem Radarbild	23
4.10	Konturbestimmung 3	24
4.11	Verschiebung der approximierenden Linien	25
4.12	Approximation mit verschiedenen geometrischen Primitiven	26

5.1	Datenstrukturen	28
5.2	Analyseergebnis als Menge von Polygonzügen	28
5.3	Varianten für den Uferverlauf	30
5.4	Karte mit Varianten für den Uferverlauf	30
5.5	Interne Darstellung der Karte	31
5.6	Aufbau eines Kalman-Filters	32
5.7	Beschreibung der Vereinigung von Polygonzug und Karte mit einem endlichen Automaten	37
5.8	Zuordnung eines Polygonzugpunktes zur Karte	38
5.9	Struktogramm für Zustand 2	39
5.10	Einzugsbereiche	40
5.11	Koordinatensysteme beim Test auf Enthaltensein eines Punktes in einer Strecke	41
5.12	Ausgangssituation im Zustand 3	42
5.13	Zustand 3, erster Fall	42
5.14	Zustand 3, zweiter Fall	43
5.15	Zustand 3, dritter Fall	43
5.16	Zustand 3, Ende der Überdeckung	44
5.17	Zustand 3, Vermeidung einer Endlosschleife	45
5.18	Zustand 5, Erkennung von Kartenpunkten, die auf der Polygonzug- strecke liegen	45
5.19	Struktogramm für Zustand 5	46
5.20	Arbeitsweise des Automaten	47
5.21	Struktogramm: Vereinigen von Kartenpunkten	48
5.22	Vereinigen von Kartenpunkten	48
5.23	Beispiel 1: Kartenausschnitt des Rheins	49
5.24	Beispiel 2: Kartenausschnitt des Rheins	50

Zusammenfassung

Am Institut für Systemdynamik und Regelungstechnik der Universität Stuttgart wird ein integriertes Navigationssystem zur automatischen Führung von Binnenschiffen entwickelt. Dieses Navigationssystem verwendet unter anderem eine elektronisch abgespeicherte Flußkarte. Diese Flußkarten werden bisher ausgehend von amtlichen Papierkarten von Hand digitalisiert. Dies ist problematisch wenn die Papierkarten veraltet oder gar nicht vorhanden sind. Deshalb ist es wünschenswert, Karten auch in anderer Weise erstellen zu können. In dieser Arbeit wird ein Verfahren zur automatischen Erstellung einer elektronischen Flußkarte aus Radarbildern beschrieben.

Da die Messungen des Radargerätes recht stark verrauscht sind, werden die Radarbilder zunächst mit einem morphologischen Filter vorverarbeitet. Anschließend werden die für eine Flußkarte relevanten Teile des Radarbildes durch Linien approximiert. Dies geschieht durch die rekursive Unterteilung einer approximierenden Linie an der Stelle ihrer maximalen Abweichung zur Kontur des Objektes. Die entstehende Darstellung ist eine Menge von Polygonzügen, die anschließend mit den Ergebnissen aus den vergangenen Zeitschritten verglichen und zu einer Karte kombiniert werden. Die Eckpunkte der Polygonzüge werden dabei mit Kalman-Filtern modelliert.

Mit dem Verfahren wurden präzise Karten von verschiedenen Rhein- und Neckarschnitten erzeugt und mit den bereits vorhandenen Karten verglichen.

Kapitel 1

Einleitung

1.1 Das Schiffsprojekt

Am Institut für Systemdynamik und Regelungstechnik (ISR) der Universität Stuttgart wird ein integriertes Navigationssystem für Binnenschiffe entwickelt [Gilles et al. 93]. Ziel dieses Projektes ist ein System, mit dessen Hilfe ein Schiff automatisch auf einer Wasserstraße geführt werden kann. Dieses Navigationssystem soll möglichst ohne zusätzliche Einrichtungen außerhalb des Schiffes auskommen. Ein solches System wird den Schiffsführer von Routinearbeiten entlasten und ihn in schwierigen Situationen, wie Nacht- oder Nebelfahrten unterstützen. Dadurch lassen sich die Sicherheit des Verkehrs auf den Wasserstraßen erhöhen und das Risiko für die Umwelt beim Transport gefährlicher Güter vermindern. Das System konnte in zahlreichen Versuchen erfolgreich erprobt werden.

Die Bezeichnung „integriertes Navigationssystem“ bedeutet, daß innerhalb des Systems Meßinformationen unterschiedlicher Sensoren mit abgespeichertem a-priori-Wissen koordiniert werden, um die Bewegung des Schiffes zu planen und auszuführen. Die eingesetzten Sensoren sind Radar und Laserscanner als bildgebende Sensoren, Kurs- oder Wendekreisler, GPS-Satellitenempfänger (GPS, Global Positioning System) zur Positions- und Geschwindigkeitsbestimmung und ein Ultraschall-Doppler-Log zur Geschwindigkeitsbestimmung gegenüber dem Grund. Zu dem a-priori-Wissen zählen ein mathematisches Modell für die Dynamik des navigierenden Schiffes, eine Reihe mathematischer Modelle zur Beschreibung der Dynamik fremder Schiffe, eine Wissensbasis für die Darstellung der Verkehrsregeln und eine elektronisch gespeicherte Karte der Wasserstraße als Modell der Flußlandschaft (siehe Abbildung 1.1). Diese Karte enthält alle sichtbaren zeitinvarianten Objekte und stellt zusätzlich die für die Schiffsführung relevanten Informationen bereit. Durch einen Bildvergleich zwischen dem neu erhaltenen Radarbild und der elektronischen Flußkarte wird mit einem Least-Squares-Verfahren die Position des navigierenden Schiffes geschätzt.

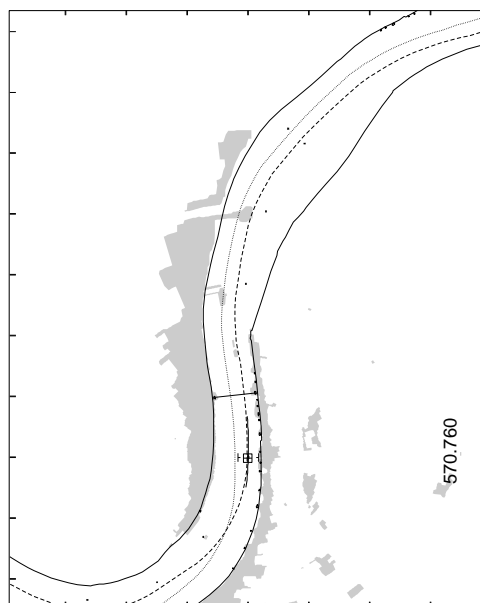


Abbildung 1.1: Elektronische Flußkarte

Für die Kommunikation mit dem Benutzer verfügt das integrierte Navigationssystem über einen Kommandointerpreter. Von den einzelnen Modulen können Kommandos zur Verfügung gestellt werden, mit deren Aufruf sich während des Programmablaufs Parameter einstellen lassen.

1.2 Aufgabenstellung

Da die Flußkarten bisher nicht in digitaler Form vorliegen, müssen sie entweder aufgrund von amtlichen Karten digitalisiert oder interaktiv durch Anklicken auf dem Radarbild erstellt werden. Die erste Vorgehensweise ist vor allem dann problematisch, wenn keine amtlichen Karten vorliegen oder bauliche Veränderungen durchgeführt werden. Aus diesem Grund sollen im Rahmen des weiteren Projektfortschritts Methoden untersucht werden, die benötigten Flußkarten automatisch zu erstellen.

Die Aufgabe umfaßt die Erstellung einer elektronischen Flußkarte direkt aus Radarbildern. Grundlage ist die aktuelle Schiffsposition, die vor allem durch DGPS (DGPS, Differential GPS) gestützt wird. Bei GPS im Differential-Mode wird ein zweiter GPS-Empfänger erforderlich, dessen Koordinaten exakt bestimmt sind. In dieser Referenzstation werden die Korrekturwerte für das GPS-Signal errechnet und dem Bordempfänger zur Korrektur seiner eigenen GPS-Meßdaten übermittelt. Während das GPS-Signal nur eine Genauigkeit von ca. 100 m erreicht, liegt die Positionsgenauigkeit bei der Verwendung von DGPS in der Größenordnung von 1-3 m.

Da die Radarbilder recht stark verrauscht sind, ist zunächst ein Filter auszuwählen, der

eine optimale Verarbeitung in den weiteren Schritten ermöglicht. Im nächsten Schritt sollen die Uferlinien mit Hilfe einfacher geometrischer Objekte angenähert werden. Die dabei entstandene Momentaufnahme muß dann mit den Beobachtungen der Vergangenheit verglichen und zu einer Karte vereinigt werden.

1.3 Übersicht

In Kapitel 2 soll zunächst ein kurzer Überblick über die aktuellen Verfahren zur Kartenerstellung autonomer Fahrzeuge gegeben werden. Es folgt Kapitel 3 über die Radarbildvorverarbeitung mit einer kurzen Einführung in die Morphologie [Gonzales et al. 92]. In Kapitel 4 wird die Vorgehensweise bei der Analyse eines Radarbildes erläutert. Dabei wird ein Algorithmus von PAUL L. ROSIN [Rosin et al. 95] vorgestellt, der Listen von verbundenen Pixeln analysiert und durch Linien annähert. Kapitel 5 zeigt, wie diese Analyseergebnisse aus aufeinanderfolgenden Zeitschritten miteinander kombiniert werden. Es folgt ein Kapitel mit einem Rückblick auf den Verlauf der Arbeit und einem Ausblick auf eventuell folgende weiterführende Arbeiten an diesem Thema.

Kapitel 2

Einführung in die Methoden der Kartenerstellung

Autonome mobile Systeme nehmen Daten aus ihrer Umwelt über Sensoren auf. Diese Daten werden so abstrahiert, daß eine Hinderniserkennung für die Navigation und eine Umweltmodellierung für die Kartenerstellung ermöglicht wird. Im folgenden soll ein Überblick über die bisher verfolgten Ansätze zur Kartenerstellung gegeben und im Hinblick auf eine Verwendung im Schiffprojekt bewertet werden.

2.1 Überblick

Es gibt zwei grundsätzliche Formalismen zur Umweltdarstellung. Zum einen die topologischen Graphen [Kurz 96]. Sie repräsentieren die Umwelt durch Graphen. Die Knoten eines Graphen korrespondieren mit verschiedenen Orten oder markanten Punkten. Zwei Knoten eines Graphen sind durch eine Kante verbunden, wenn die entsprechenden Orte unmittelbar durch einen Weg verbunden sind. Solche topologischen Graphen repräsentieren die Umwelt sehr kompakt aber auch detailarm. Sie werden hauptsächlich für eine globale Routenplanung benutzt, denn hier ist eine Planung auf hoher Abstraktionsebene erforderlich. Im Gegensatz dazu stehen die geometrischen Umweltdarstellungen. Sie eignen sich besser für eine lokale Planung, da sie die Umgebung detailreicher darstellen. Die geometrischen Modelle lassen sich ihrerseits in zwei nahezu komplementäre Typen unterteilen: gitterbasierte [Kanesalingam et al. 93], [Stuck et al. 94] und merkmalsbasierte Ansätze [Borthwick et al. 94], [Kurz et al. 92].

Bei gitterbasierten Ansätzen wird die Umgebung auf ein Gitter von Zellen abgebildet. Jeder Zelle wird ein Wert zugewiesen, der die Wahrscheinlichkeit, daß sich in dieser Zelle ein Hindernis befindet, widerspiegelt. Der Update-Algorithmus einer solchen Karte läuft typischerweise in folgenden Schritten ab. Ausgehend von der Position des

Sensors wird eine Linie entlang der Achse des Sensorstrahles auf das Kartengitter projiziert. Dann wird berechnet, welche Zellen des Gitters innerhalb eines gewissen Bereiches um diese Linie liegen. Der Algorithmus erhöht dann die Werte in den Zellen um den Endpunkt der Linie. Dieser Endpunkt korrespondiert mit einem Objekt, das von dem Sensor erkannt wurde. Zuletzt werden die Wahrscheinlichkeiten in den zuvor berechneten Zellen um die Linie erniedrigt, da hier in der letzten Messung kein Hindernis erkannt wurde.

Die merkmalsbasierten Ansätze modellieren Objekte oder Teile von Objekten der Umgebung mit geometrischen Primitiven und assoziieren Messungen mit ihnen. Der Update der Karte findet also durch explizites Vergleichen von Merkmalen des Sensors mit den Merkmalen der Karte statt.

In Abbildung 2.1 werden die möglichen Ergebnisse einer Kartengenerierung mit den verschiedenen Ansätzen aufgezeigt. 2.1 (a) ist das Original. 2.1 (b) zeigt das Ergebnis einer gitterbasierten Kartenerstellung. Die Helligkeit der Gitterpunkte gibt an, mit welcher Sicherheit in den entsprechenden Zellen ein Hindernis erwartet wird. In 2.1 (c) ist das Ergebnis des merkmalsbasierten Ansatzes zu sehen. Die Kontur wurde mit Linien dargestellt. Ein möglicher topologischer Graph für die Umgebung ist 2.1 (d).

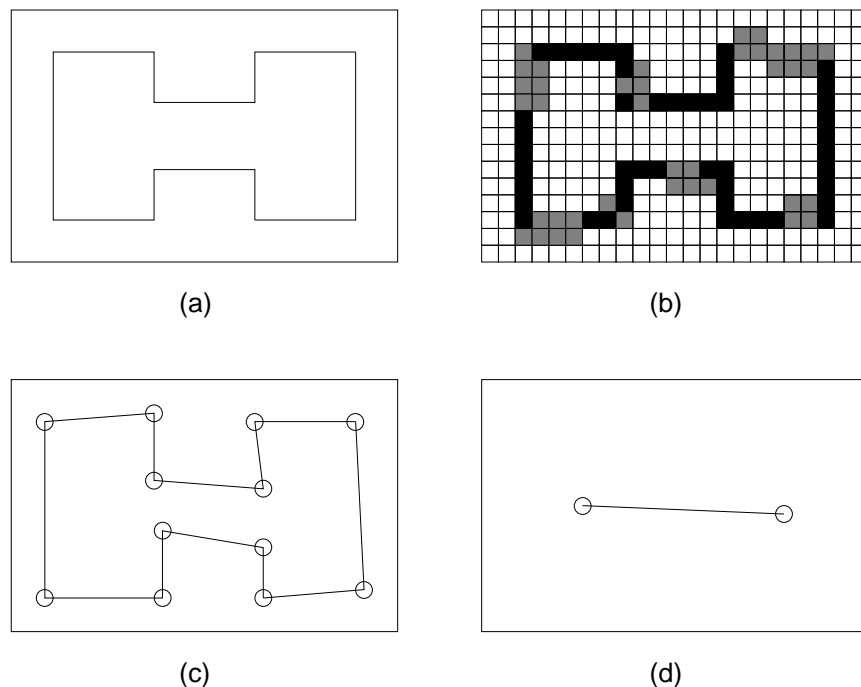


Abbildung 2.1: Arten der Umweltrepräsentation

Es wurden auch Kombinationen der verschiedenen Ansätze erprobt. In [Feiten et al. 92] wird gleichzeitig eine gitterbasierte und eine merkmalsbasierte Karte verwendet. Ein Beispiel für die Kombination eines topologischen Graphen mit einer gitterbasierten Karte ist in [Thrun et al. 96] nachzulesen.

2.2 Bewertung

Der Aufbau der elektronischen Flußkarte orientiert sich an den Anforderungen, die sie im integrierten Navigationssystem erfüllen muß. Da einerseits von der Karte zahlreiche Details für die Schiffsführung benötigt werden und andererseits keine globale Routenplanung stattfindet, ist die Darstellung als topologischer Graph ungeeignet. Der gitterbasierte Ansatz leidet unter seiner großen Platzkomplexität. Um alle wichtigen Details der Umwelt zu erfassen, müssen die Zellen des Gitters hinreichend klein gewählt werden. Bei einer Zellengröße von einem Meter müßten also für einen Quadratkilometer bereits eine Million Zellen abgespeichert werden. Außerdem würde die rechteckige Form eines Gitters sehr viele überflüssige Informationen enthalten. Der Extremfall wäre ein Flußverlauf diagonal durch ein quadratisches Gitter. Der Platzbedarf des merkmalsbasierten Ansatzes ist vergleichsweise gering, da hier die Umgebung durch relativ wenige geometrische Primitive beschrieben werden kann. Ein weiterer Vorteil dieser Darstellung ist die Nähe zur Datenstruktur der bestehenden Flußkarten (siehe [Neul 92]). Dadurch wird eine spätere Anpassung der automatisch generierten Flußkarten an die Anforderungen des integrierten Navigationssystems vereinfacht.

Kapitel 3

Radarbildvorverarbeitung

3.1 Einführung in die Morphologie

Das Wort Morphologie bezeichnet gewöhnlich einen Zweig der Biologie, der sich mit der Form und Struktur von Pflanzen befaßt. Im mathematischen Kontext ist die Morphologie ein Werkzeug zur Verarbeitung von Bildern [Gonzales et al. 92]. Sie ermöglicht dabei z.B. die Glättung der Konturen von Objekten, wenn diese bei der Aufnahme durch Sensoren verrauscht wurden. Die Sprache der mathematischen Morphologie ist die Mengentheorie. Damit können die meisten Probleme der Bildverarbeitung angegangen werden. Die Mengen repräsentieren die Formen der Objekte in einem Bild. Beispielsweise ist ein Binärbild vollständig durch die Menge der im Bild gesetzten Pixel beschrieben. Die Elemente der Menge sind in diesem Fall zweidimensionale Vektoren aus dem ganzzahligen Raum $\mathcal{Z} \times \mathcal{Z}$. Ein Pixel ist also repräsentiert durch seine Koordinaten (x, y) .

3.1.1 Dilation und Erosion

Hier werden zwei grundlegende morphologische Operationen vorgestellt. Sie bilden die Basis für das *Opening* und *Closing*, die im nächsten Abschnitt beschrieben werden.

Vorbereitende Definitionen

Definition 3.1 (Translation) *A und B seien Mengen in $\mathcal{Z} \times \mathcal{Z}$ mit den Elementen $a = (a_1, a_2)$ und $b = (b_1, b_2)$. Dann sei die Translation von A mit $x = (x_1, x_2)$, geschrieben $(A)_x$, definiert als*

$$(A)_x = \{c \mid c = a + x, \forall a \in A\}.$$

Definition 3.2 (Spiegelung) Die Spiegelung von B , geschrieben \hat{B} , ist definiert als

$$\hat{B} = \{x | x = -b, \forall b \in B\}.$$

Definition 3.3 (Komplement) Das Komplement einer Menge A , geschrieben A^c , ist definiert als

$$A^c = \{x | x \notin A\}.$$

Definition 3.4 (Differenz) Die Differenz zweier Mengen A und B , geschrieben $A - B$, ist definiert als

$$A - B = \{x | x \in A, x \notin B\} = A \cap B^c.$$

Damit sind die Voraussetzungen für die Definition der Begriffe *Dilation* und *Erosion* geschaffen.

Definition 3.5 (Dilation) Seien A und B Mengen in $\mathcal{Z} \times \mathcal{Z}$ und \emptyset die leere Menge. Dann ist die Dilation von A mit B , geschrieben $A \oplus B$, definiert als

$$A \oplus B = \{x | (\hat{B})_x \cap A \neq \emptyset\}.$$

Eine Dilation von A mit B besteht somit aus einer Spiegelung von B am Ursprung, gefolgt von einer Verschiebung dieser Spiegelung um den Vektor x . Die Dilation von A mit B ist dann die Menge aller x , so daß \hat{B} nach dieser Verschiebung A an mindestens einem Punkt überlappt.

Definition 3.6 (Erosion) Seien A und B Mengen in $\mathcal{Z} \times \mathcal{Z}$. Dann ist die Erosion von A mit B , geschrieben $A \ominus B$, definiert als

$$A \ominus B = \{x | (B)_x \subseteq A\}.$$

Folglich ist eine Erosion von A mit B die Menge aller Punkte x , so daß das um x verschobene B komplett in A enthalten ist.

In diesen morphologischen Operationen wird die Menge B als das *Strukturelement* bezeichnet.

3.1.2 Opening und Closing

Zwei weitere wichtige morphologische Operationen sind das *Opening* und das *Closing*. Sie setzen sich jeweils aus einer Erosion und einer Dilation zusammen und unterscheiden sich lediglich in der Reihenfolge der Ausführung dieser beiden Operationen. Daraus ergeben sich folgende Definitionen:

Definition 3.7 (Opening) *Das Opening einer Menge A mit einem Strukturelement B , geschrieben $A \circ B$, ist definiert als*

$$A \circ B = (A \ominus B) \oplus B$$

Definition 3.8 (Closing) *Das Closing einer Menge A mit einem Strukturelement B , geschrieben $A \bullet B$, ist definiert als*

$$A \bullet B = (A \oplus B) \ominus B$$

Das Opening glättet die Kontur eines Objektes, bricht schmale Verbindungen auf und beseitigt aus dem Objekt hervorstehende dünne Gebiete. Das Closing glättet ebenfalls die Konturen, verstärkt aber schmale Verbindungen, läßt schmale Risse verschwinden, überdeckt kleine Löcher im Inneren des Objektes und schließt Lücken in der Kontur.

Für Opening und Closing gibt es einfache und anschauliche geometrische Interpretationen. Das Ergebnis des Openings entsteht durch eine Verschiebung des Strukturelementes im Inneren des Originalbildes, so daß immer das ganze Strukturelement im Originalbild enthalten ist. Alle dabei vom Strukturelement überdeckten Punkte bilden dann die resultierende Gestalt.

Das Ergebnis des Closings entsteht durch eine Verschiebung des Strukturelements außerhalb des Originalbildes, so daß sich Originalbild und Strukturelement nicht überlappen. Die Punkte, die damit nicht überdeckt wurden, bilden hier die resultierende Gestalt.

Aus den Opening- und Closing-Operationen kann nun ein morphologischer Filter konstruiert werden. Das Ziel eines solchen Filters ist es, das Hintergrundrauschen im Bild und die Auswirkungen des Rauschens auf das Objekt zu eliminieren (siehe Abbildung 3.1). Gleichzeitig soll aber das Objekt selbst so wenig wie möglich verändert werden.

Ein solcher Filter wurde im Rahmen des Schiffsprojektes bereits implementiert. Er wird im Abschnitt 3.3 vorgestellt.

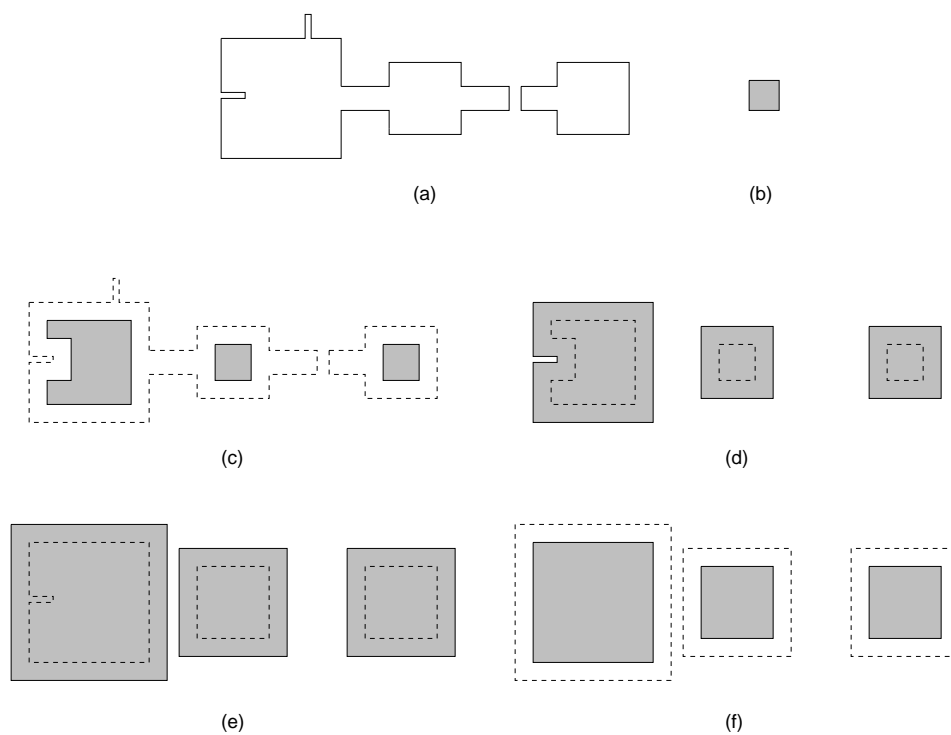


Abbildung 3.1: Morphologischer Filter: (a) Originalbild A ; (b) Strukturelement B ; (c) Erosion $A \ominus B$; (d) Ergebnis des Openings $A \circ B = (A \ominus B) \oplus B$; (e) Ergebnis nach Dilation auf das Opening $(A \circ B) \oplus B$; (f) Endergebnis nach Opening und Closing $[(A \circ B) \oplus B] \ominus B = (A \circ B) \bullet B$.

3.2 Das Radarbild

Ein Standardradargerät, wie es auf den meisten Binnenschiffen bereits existiert, liefert die Radarbilder. Eine detaillierte Beschreibung ist in [Neul 92] nachzulesen. Es handelt sich dabei um ein Rundsichtgerät, dessen Antenne sich in ca. 2,3 Sekunden einmal um seine Hochachse dreht. Die Radarimpulse werden mit einer Frequenz von 3 kHz ausgesendet, so daß das gesamte Radarbild aus ca. 6900 radialen Strahlen besteht. Der Winkelabstand zweier Strahlen ist somit ca. $0,05^\circ$. Weil die Auflösung der Radarantenne in azimuthaler Richtung ungefähr 1° beträgt, enthält diese Darstellung redundante Informationen. Ein digitaler Vorfilter faßt die Echoinformationen so zusammen, daß ein Radarbild mit 720 Strahlen zur Verfügung steht. Anschließend wird dieses Bild in der *Run-Length-Codierung* dargestellt, d.h. von jeder Echostrecke werden jeweils nur Start- und Endpunkt abgespeichert. Diese Darstellung nutzt die Tatsache, daß Radarechos in den meisten Fällen als größere Gebiete erscheinen. Das legt die in Abbildung 3.2 beschriebene Datenstruktur nahe.

0-ter Strahl	1. Echo		...	m-tes Echo	
	start	stop	...	start	stop
...					
(n-1)-ter Strahl	1. Echo		...	m-tes Echo	
	start	stop	...	start	stop

Abbildung 3.2: Datenstruktur eines Radarbildes

Die Strahlen liegen in einem Array von 0 bis 719. Jeder Strahl ist ein Array von 0 bis (MAX_ECHO-1). MAX_ECHO ist die Zahl der maximal in einem Radarstrahl vorkommenden Echos. Jedes Echo ist wiederum ein Verbund aus den Elementen *start* und *stop*, die die Abstände der Anfangs- und Endpunkte der Echostrecke enthalten. Die Entfernungen haben dabei die Einheit „RUnit“ (i.a. 1 RUnit = 3 m), die der Auflösung des Radarstrahles in radialer Richtung entspricht. Jedes Echo ist also durch Strahlenindex und Echoindex eindeutig bestimmt.

3.3 Der Filter *morph*

Der am Schiffsprojekt implementierte morphologische Filter *morph* arbeitet aufgrund der in Abschnitt 3.2 beschriebenen Datenstruktur des Radarbildes nicht auf kartesischen, sondern auf Polarkoordinaten. Weil bei steigender Entfernung vom Ursprung des Polarkoordinatensystems der Abstand zwischen zwei Radarstrahlen zunimmt und der Meßfehler des Radargeräts größer wird, muß die Ausdehnung des Strukturelementes in azimuthaler Richtung angepaßt werden. Die Gestalt des Strukturelementes wird

deshalb durch drei Parameter bestimmt: zum einen durch die Konstante c und den entfernungsabhängigen Wert $w(r)$, die zusammen die Größe in azimuthaler Richtung $\Delta w(r)$ bilden und zum anderen durch die Größe in radialer Richtung Δr . Dem Algorithmus *morph* werden deshalb drei Parameter zur Festlegung des Strukturelementes übergeben (siehe Abbildung 3.3).

Radialer Fehler [RUnit] =
 Konstanter azimuthaler Fehler [m] =
 Entfernungsabhängiger azimuthaler Fehler [Grad] =

Abbildung 3.3: Parameter für den *morph*-Aufruf

Für *opening* und *closing* ist die Einstellung unterschiedlich großer Strukturelemente möglich.

Der entfernungsabhängige Fehler in azimuthaler Richtung wird in Form des Winkels β eingegeben. Für $w(r)$ gilt dann

$$w(r) = r \tan \beta \quad (3.1)$$

Wegen $\sin \alpha \approx \alpha$ für kleine α und $\cos \alpha \approx 1$ für kleine α gilt mit guter Näherung (Fehler $< 0,1\%$)

$$w(r) = r \tan \beta = r \frac{\sin \beta}{\cos \beta} \approx r \beta \quad (3.2)$$

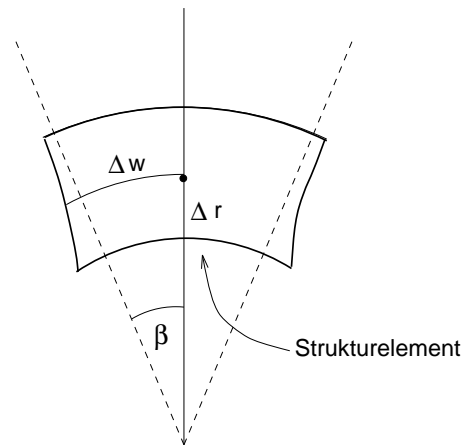
Die resultierende Größe für den azimuthalen Fehler $\Delta w(r)$ wird dann gemäß

$$\Delta w = \sqrt{c^2 + w(r)^2} \quad (3.3)$$

bestimmt. Dadurch dominiert der konstante Fehler c für kleine Entfernungen r vom Ursprung und der entfernungsabhängige Fehler $w(r)$ für große r . In Abbildung 3.4 ist die besondere Form des Strukturelementes dargestellt.

3.4 Anpassung an die Anforderungen der Kartengenerierung

Für die Kartengenerierung ist es wünschenswert, daß die Radarbilder so vorverarbeitet werden, daß die Konturen der Objekte glatter, auftretende Lücken in den Uferlinien geschlossen und sehr kleine Objekte, die auf das Meßrauschen zurückzuführen sind,

Abbildung 3.4: Strukturelement des Filters *morph*

von vornherein ausgeschlossen werden. Die Abbildung 3.5 enthält Beispiele für Ergebnisse der morphologischen Filterung mit verschiedenen Parametern. Links oben ist das Originalbild vor der Filterung.

In Kapitel 4 wird ein Algorithmus vorgestellt, der die Konturen der Radarbildobjekte durch Linien annähert. Die Anzahl der Linien, die dieser Algorithmus bei ein- und ausgeschalteter morphologischer Vorverarbeitung in die Uferabschnitte legt, ist in Tabelle 3.1 gegenübergestellt. Es wurde dabei eine Sequenz von zehn Radarbildern approximiert. Die Ergebnisse zeigen, daß die glatteren Konturen nach der morphologischen Vorverarbeitung durch weniger Linien dargestellt werden können.

Bild	1	2	3	4	5	6	7	8	9	10
oM	16	23	37	29	24	46	24	25	30	22
mM	14	17	22	20	15	20	18	19	25	23

Tabelle 3.1: Linienzahl mit (mM) und ohne morphologische Vorverarbeitung (oM).

Als recht gute Werte für die weitere Verarbeitung in den folgenden Schritten haben sich

$$\begin{aligned}\Delta r &= 2 \text{ [RUnit]} \\ c &= 5 \text{ m} \\ \beta &= 3^\circ\end{aligned}$$

für das *closing* und

$$\Delta r = 1 \text{ [RUnit]}$$

$$c = 2 \text{ m}$$

$$\beta = 0.5^\circ$$

für das *opening* herausgestellt.

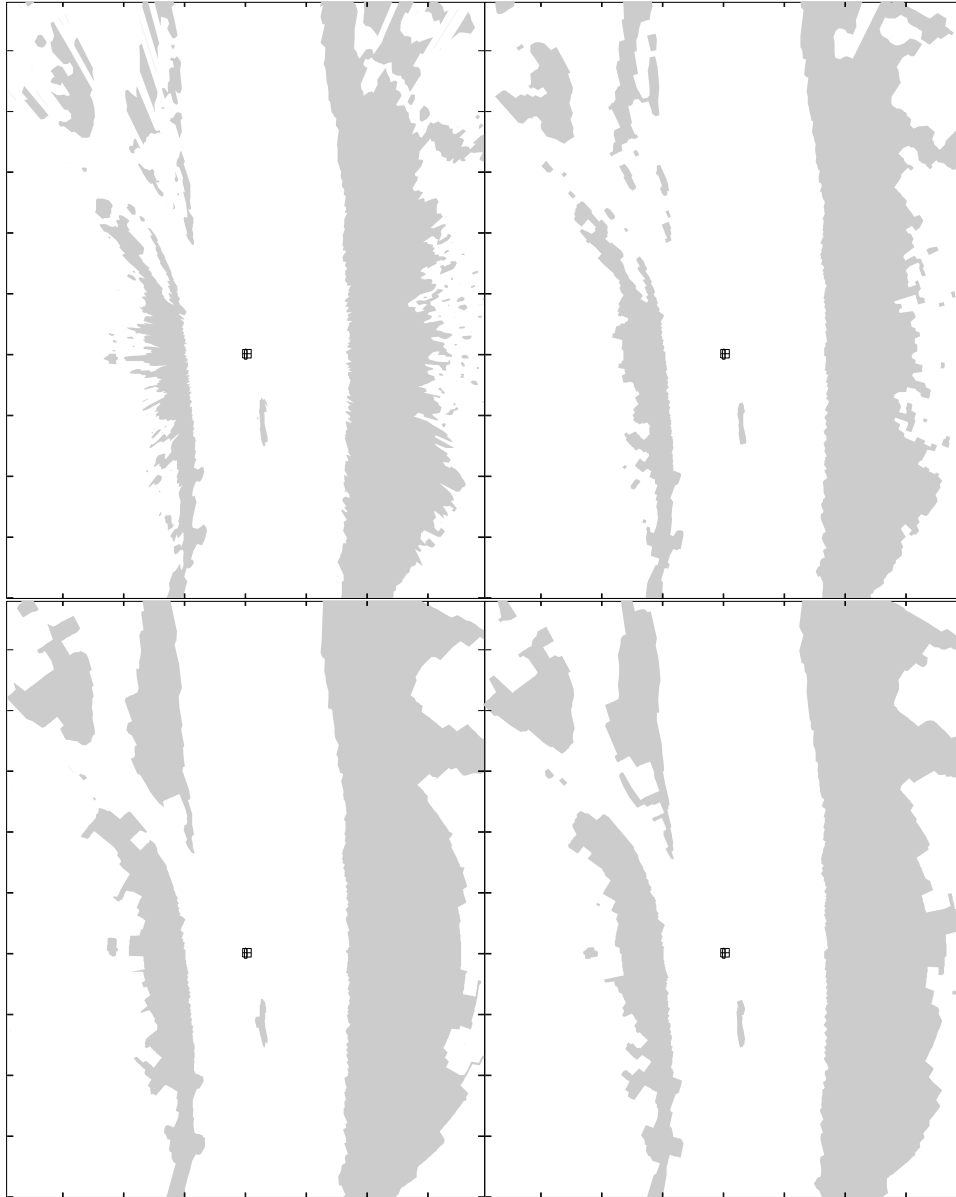


Abbildung 3.5: Links oben: Radarbild ohne Vorfilterung; Rechts oben: Radarbild nach morphologischer Filterung, Parameter $\Delta r = 1, c = 3, \beta = 0.7$ (Closing) und $\Delta r = 1, c = 3, \beta = 0.7$ (Opening), Links unten: $\Delta r = 2, c = 5, \beta = 3$ und $\Delta r = 0, c = 0, \beta = 0$, Rechts unten: $\Delta r = 2, c = 5, \beta = 3$ und $\Delta r = 1, c = 2, \beta = 0.5$.

Kapitel 4

Analyse eines Radarbildes

Gegenstand dieses Kapitels ist die Analyse des Radarbildes, das durch den im vorangegangenen Kapitel beschriebenen Filterprozeß erhalten wird. Am Ende der Analyse soll eine Darstellung des Ufers in Form einer Menge von Polygonzügen stehen, die die Konturen aus dem Radarbild annähern. Diese Polygonzüge sind dann die Eingabe für den in Kapitel 5 beschriebenen Algorithmus zur Vereinigung der Analyseergebnisse aus aufeinanderfolgenden Zeitschritten.

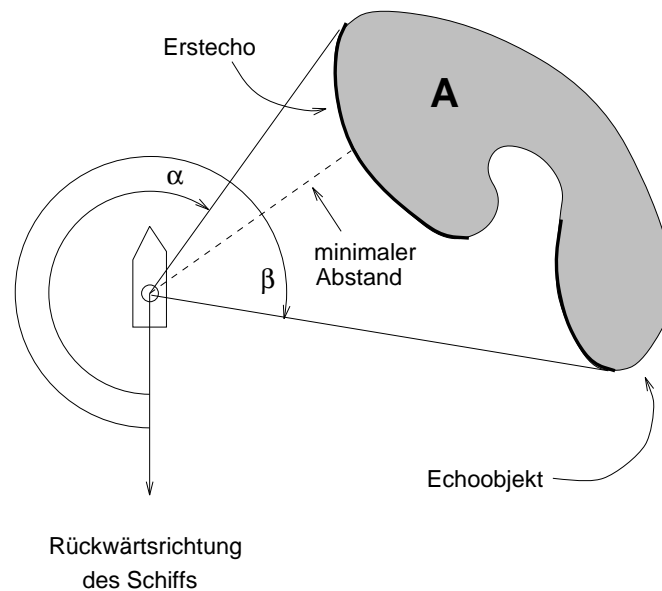
Für die Interpretation des Inhalts von Radarbildern ist es notwendig, die in Abschnitt 3.2 beschriebene bildpunkt-basierte Datenrepräsentation zu verlassen und zu einer objektorientierten Beschreibung des Radarbildinhalts überzugehen [Hermann 94], [Neul 92]. Denn nur so läßt sich das Radarbild sinnvoll auswerten.

Zunächst wird das Radarbild in zusammenhängende Echogebiete, in *Objekte*, eingeteilt. Dies geschieht durch den Vergleich der Echos zweier aufeinanderfolgender Radarstrahlen. Im Falle einer Überlappung oder Berührung in radialer Richtung werden die beiden Echos dem gleichen Echoobjekt zugeordnet.

Im nächsten Schritt werden verschiedene Objektmerkmale aus den berechneten Echoobjekten extrahiert (siehe Abbildung 4.1). Dazu gehören

- Die Fläche A der Objekte.
- Der Erstechoanteil der Objekte. Das ist der Teil der Kontur eines Objekts, der, von der Radarantenne aus gesehen, nicht verdeckt ist.
- Minimal- und Maximalwert des Winkels der Objekte.
- Minimale Entfernung des Objekts zur Radarantenne.

Diese Informationen über die Unterteilung des Radarbildes in Objekte und die Objektmerkmale werden im integrierten Navigationssystem in Form einer *Objekttabelle* gespeichert.



α ... minimaler Sichtwinkel

β ... maximaler Sichtwinkel

Abbildung 4.1: Objektmerkmale

4.1 Auswahl relevanter Objekte

Aufgrund der oben beschriebenen Objekttable mit den darin enthaltenen Objektmerkmalen ist es möglich, die für die Kartengenerierung relevanten Objekte auszuwählen. Dies geschieht mit Hilfe einfacher Schwellwertkriterien für den Erstechoanteil der Kontur, die Objektfläche und die Entfernung des Objektes. Die Schwellwerte lassen sich im Kommandointerpreter beliebig einstellen (siehe Anhang A). Voreingestellt sind die Werte

```

min_sichtbar = 0.1 %
min_objekt_flaeche = 10000 m2
max_distanz = 100 RUnit.

```

Durch den Schwellwert für den Erstechoanteil werden Objekte verworfen, die fast komplett hinter dem Ufer liegen und nur an einzelnen Stellen direkt sichtbar sind (siehe Beispiel in Abbildung 4.2).

Der Schwellwert für die Objektfläche stellt sicher, daß andere Schiffe, die ebenfalls in der Objekttable eingetragen sind, erkannt und aussortiert werden. Die untere Schranke für die Objektdistanz verwirft Objekte, die sehr weit vom Radargerät entfernt und dadurch ungenau sind.

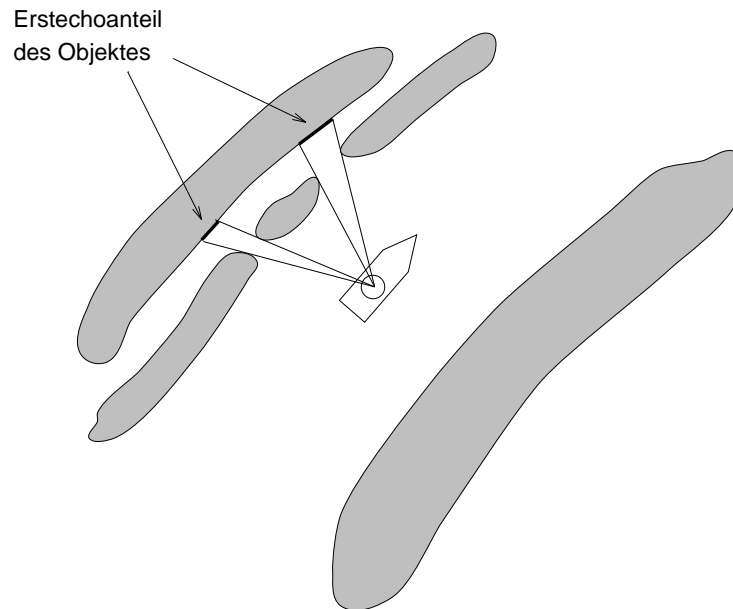


Abbildung 4.2: Objektauswahl, Beispiel Erstechoanteil

Nachdem die Objekte, die in die Kartenerstellung eingehen sollen, ausgewählt sind, wird deren Kontur bestimmt. Hiervon soll nur der Teil berücksichtigt werden, der dem Fluß zugewandt ist. Dies geschieht durch einen Aufruf der Prozedur *kontur*, die die einzelnen Konturpunkte in Polarkoordinaten bestimmt. Anschließend werden die Konturen von den Rändern aus so gekürzt, daß die beiden Endpunkte der Kontur ein Erstecho sind. Durch diese Vorgehensweise bleiben Ränder von Konturen unberücksichtigt, die im hinteren Teil des Radarbildes verlaufen und nicht mehr zur Uferlinie zählen (siehe Beispiel in Abbildung 4.3).

Gleichzeitig bleiben aber Teile der Kontur erhalten, die zwar nicht direkt sichtbar, aber von sichtbaren Teilen der Kontur eingeschlossen sind (siehe Beispiel in Abbildung 4.4).

4.2 Approximation der Objektkonturen durch Linien

Die Konturpunkte, die nach einem *kontur*-Aufruf in Polarkoordinaten vorliegen, werden anschließend in Gauß-Krüger-Koordinaten umgerechnet und in einer Pixelliste, die aus zwei Feldern für die x - und y -Koordinaten besteht, abgelegt.

Der Algorithmus von PAUL L. ROSIN [Rosin et al. 95] wählt aus diesen Punkten Eckpunkte für Strecken aus, die die Kontur annähern. In [Knieriemen et al. 90] wird ein Algorithmus zur Erkennung von Ecken in den Umweltkonturen vorgestellt, der auf

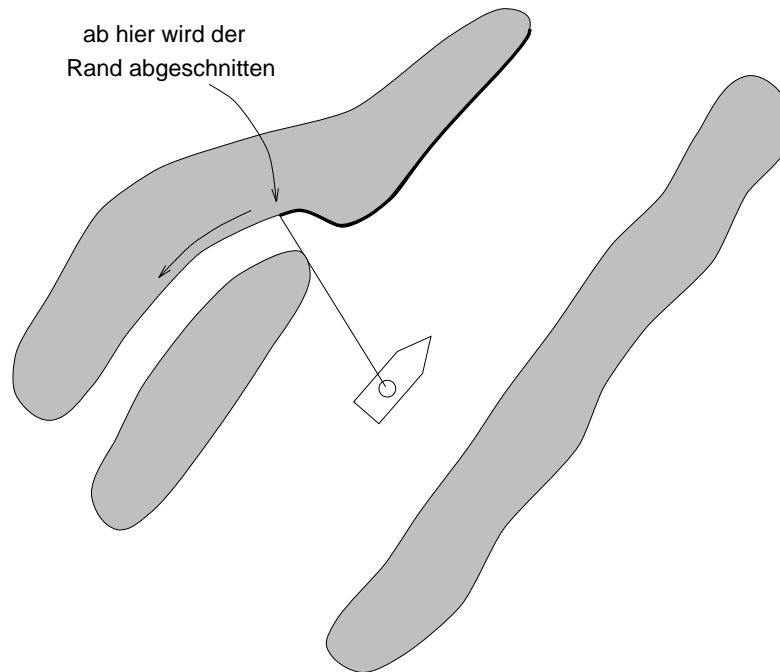


Abbildung 4.3: Konturbestimmung, Abschneiden unwichtiger Ränder

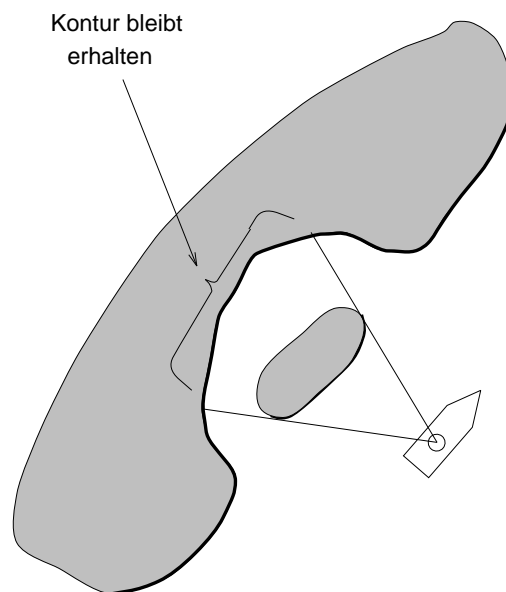


Abbildung 4.4: Konturbestimmung, Erhaltung nur streckenweise verdeckter Konturteile

der Analyse einer *Krümmungsfunktion* basiert. Die Krümmung in einem Meßpunkt ergibt sich dabei aus der Differenz der Orientierung der Strecken zum letzten und dem nächsten Meßpunkt. Ein Peak in der Krümmungsfunktion entspricht dann einer Ecke. Mit dem *Eckenfilter* ist in [Kurz et al. 92] ein weiteres Verfahren nachzulesen.

In dieser Arbeit wird der Algorithmus von PAUL L. ROSIN verwendet. Seine Vorteile sind die einfache Implementierung und die Erweiterbarkeit auf andere Objektarten, wie zum Beispiel Ellipsen und Spiralen.

Der Algorithmus nähert zunächst die gesamte Kontur durch eine Linie an. Der Startpunkt der Linie ist der erste Punkt der Kontur, der Endpunkt der letzte. Dann wird die Pixelliste an dem Punkt der maximalen Abweichung von Kontur und Linie in zwei Teile aufgeteilt (siehe Abbildung 4.5). Dieser Prozeß von Approximation und Unterteilung wird rekursiv für beide Teile der Liste fortgesetzt. Eine Pixelliste wird nicht mehr weiter unterteilt, wenn die maximale Abweichung von Pixelliste zu Linie unter einer vorgegebenen Schranke m bleibt. Diese Schranke läßt sich im Kommandointerpreter durch den Befehl `approx` verändern. Durch diese Vorgehensweise entsteht eine Repräsentation der Pixelliste in Form eines Binärbaums (siehe Abbildung 4.5 rechts unten).

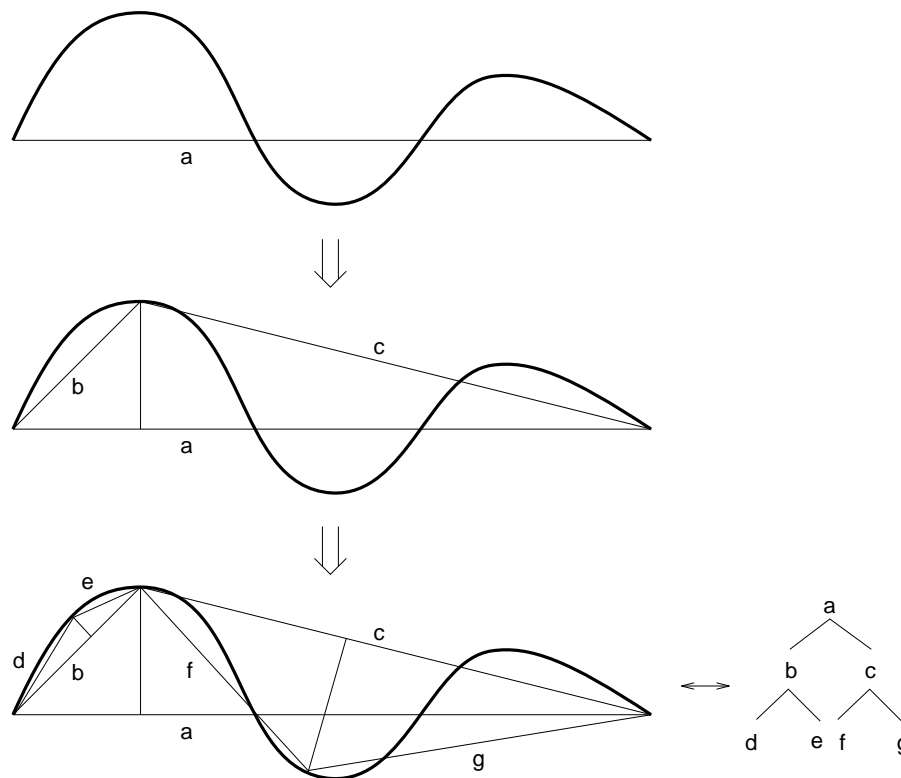


Abbildung 4.5: Annäherung einer Kurve durch Linien und Darstellung der Unterteilung als Binärbaum.

Darin beschreibt jede Stufe die komplette Kurve, jedoch von oben nach unten mit ansteigender Genauigkeit.

Jedem Knoten des Baumes, und damit jeder Linie in der Approximation, wird im Laufe der Berechnung ein *significance*-Wert zugewiesen. Dieser Wert berechnet sich als

$$\textit{significance} = \frac{\text{maximale Abweichung Linie} \leftrightarrow \text{Kurve}}{\text{Länge der Linie}}. \quad (4.1)$$

Sobald die beiden Unterbäume eines Knotens bestimmt sind, wird die *significance* des Knotens mit den *significance*-Werten der beiden Sohnknoten verglichen. Hat der Vaterknoten eine kleinere *significance* als beide Sohnknoten, so wird die Unterteilung an dieser Stelle rückgängig gemacht. Anderenfalls bleibt die Beschreibung durch die beiden Sohnknoten erhalten. In Abbildung 4.6 wird diese Vorgehensweise an einem Beispiel verdeutlicht.

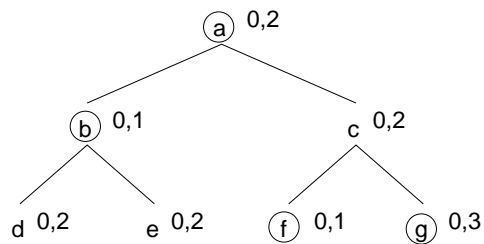


Abbildung 4.6: Auswahl einer Repräsentation anhand der *significance*-Werte

Der Knoten *f* hat einen kleineren *significance*-Wert als *c*. Die Unterteilung von *c* in *f* und *g* bleibt deshalb erhalten. Die *significance*-Werte von *d* und *e* sind beide größer als der von *b*. Die Unterteilung von *b* in *d* und *e* wird deshalb rückgängig gemacht. Die Unterteilung von *a* in *b* und *c* bleibt erhalten, da der *significance*-Wert von *b* kleiner ist, als der von *a*. Die resultierende Beschreibung besteht damit aus den Linien *b*, *f* und *g*.

Die Bestimmung der *significance*-Werte nach Gleichung 4.1 und das Überarbeiten des Binärbaumes mit Hilfe dieser Werte ist eine heuristische Vorgehensweise. Es ist nicht unmittelbar einzusehen, ob die Anwendung dieses Kriteriums sinnvoll ist und eine Verbesserung des Ergebnisses liefert. In Abbildung 4.7 sind Beispiele für die Wirkungsweise des Kriteriums zu sehen. Im Fall 4.7 (a) bleibt die Unterteilung erhalten. In 4.7 (b) wird die Unterteilung von *a* in *b* und *c* allerdings wieder verworfen, weil *a* einen kleineren Wert als *b* und *c* hat. Im Fall 4.7 (c) bleibt die Unterteilung erhalten, da *a* den größeren *significance*-Wert hat. Wie in den Fällen 4.7 (b) und (c) zu sehen ist, leidet das Kriterium darunter, daß es nur die maximale Abweichung anstelle der Fläche zwischen Konturkurve und Approximation verwendet.

Um das Ergebnis nach der Überarbeitung mit dem *significance*-Kriterium einordnen zu können, werden in Tabelle 4.1 die quadratischen Fehler der Approximation mit und

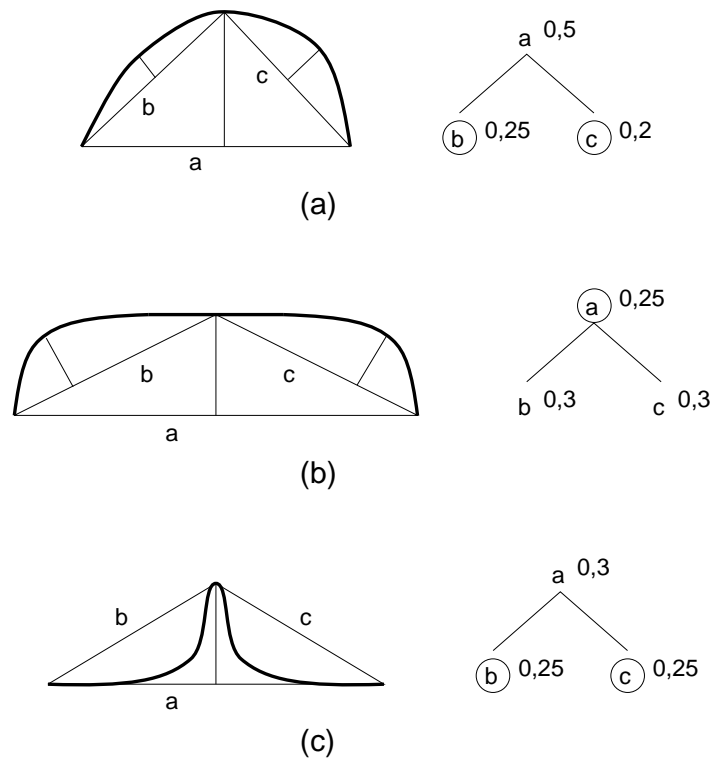


Abbildung 4.7: Das „significance-Kriterium“

ohne *significance*-Kriterium gegenübergestellt. Die Eingabedaten sind Konturpunkte eines Ufers aus zehn aufeinanderfolgenden Radarbildbildern.

Bild		1	2	3	4	5	6	7	8	9	10
Pixel		383	375	385	399	369	363	368	375	435	410
MS	<i>k</i>	39	38	30	33	28	22	21	24	36	29
	<i>F</i>	203	178	278	435	205	240	359	321	417	386
	\bar{F}	0,379	0,383	0,557	0,778	0,469	0,587	0,835	0,721	0,637	0,788
OS	<i>k</i>	39	38	33	39	29	28	27	31	39	36
	<i>F</i>	203	178	285	294	204	189	252	198	308	212
	\bar{F}	0,379	0,383	0,570	0,522	0,465	0,454	0,578	0,439	0,469	0,426

Tabelle 4.1: Fehler der Approximation. MS = mit *significance*-Kriterium, OS = ohne *significance*-Kriterium. *k* = Anzahl Linien, *F* = quadratischer Fehler, \bar{F} = mittlerer quadratischer Fehler.

Um die Berechnungen zu vereinfachen, wird der Ursprung des Koordinatensystems jeweils in den Startpunkt der Linie verschoben und die *x*-Achse in Richtung der Linie gedreht (siehe Abbildung 4.8). In diesem Koordinatensystem sei das Konturkurven-

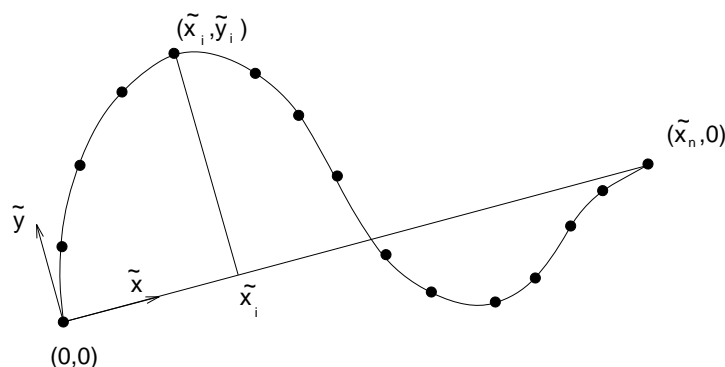


Abbildung 4.8: Konturpunkte und approximierende Linie

segment K_i durch die n Punkte $\{(\tilde{x}_l, \tilde{y}_l) \mid l = 1 \dots n\}$ gegeben. Die Endpunkte der Linie sind dann $(0,0)$ und $(\tilde{x}_n, 0)$. Der quadratische Fehler F_i zwischen einer approximierenden Linie und dem dazugehörigen Konturkurvenssegment wird hier berechnet als

$$F_i = \sum_{l=2}^n \tilde{y}_{l-1}^2 (\tilde{x}_l - \tilde{x}_{l-1}). \quad (4.2)$$

Der gesamte quadratische Fehler F aller k Segmente ist dann

$$F = \sum_{i=1}^k F_i. \quad (4.3)$$

Der mittlere quadratische Fehler \bar{F} ergibt sich schließlich zu

$$\bar{F} = \frac{F}{l_{\text{ges}}} \quad \text{mit} \quad l_{\text{ges}} = \sum_{i=1}^{k-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \quad (4.4)$$

wobei x und y jetzt wieder als Koordinaten im globalen Koordinatensystem zu verstehen sind.

Bei den zehn Testbildern wurde in der Approximation ohne das *significance*-Kriterium das Ufer mit durchschnittlich 33,9 Linien angenähert. Mit dem *significance*-Kriterium wird diese Zahl um 11,5 % auf 30 Linien verringert. Dieser Effekt wird allerdings durch einen großen Fehler der Approximation erkauft. Der mittlere Fehler steigt von durchschnittlich 0,6845 um 14,4 % auf 0,7832 an.

Abbildung 4.9 enthält Beispiele für die Approximation der Uferlinien mit diesem Algorithmus. Die Konturen der Objekte wurden wie in Abschnitt 4.1 beschrieben ausgewählt und nacheinander verarbeitet.



Abbildung 4.9: Approximation der Uferlinien in einem Radarbild

4.3 Bewertung der Ergebnisse

Anstatt der in Abschnitt 4.1 beschriebenen Bearbeitung der Objektkonturen sind natürlich auch andere Vorgehensweisen denkbar. So könnte man zum Beispiel von den Radarstrahlen jeweils nur das erste Echo berücksichtigen. Dadurch ließen sich die erhaltenen Konturen in manchen Fällen vereinfachen (siehe Abbildung 4.10 links).

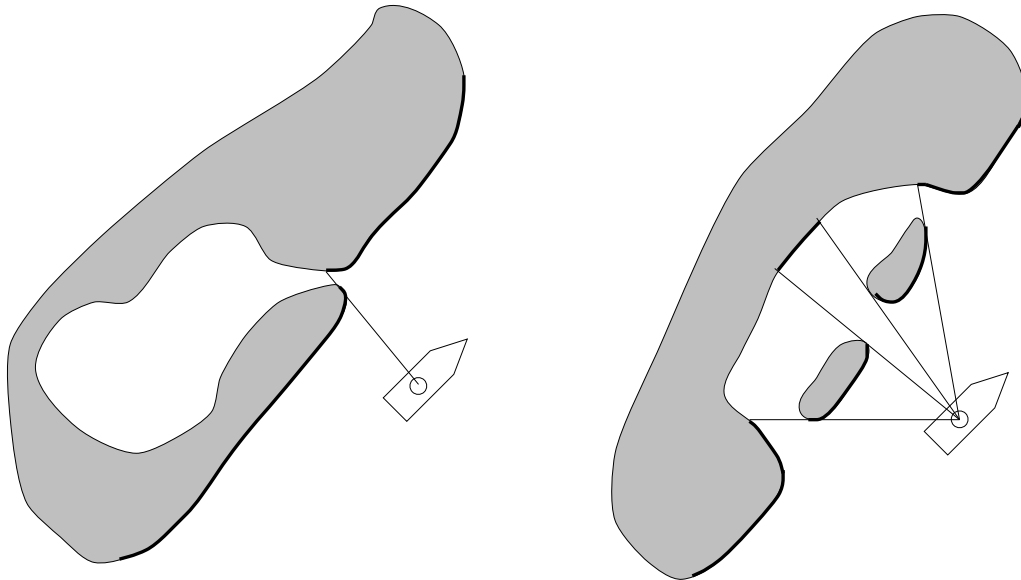


Abbildung 4.10: Konturbestimmung, nur Erstechos berücksichtigen

Da aber in diesem Fall jede Kontur unterbrochen wird, sobald sie von einem Objekt verdeckt wird, steigt dadurch die Zahl der Konturen an (siehe Abbildung 4.10 rechts).

Der in Abschnitt 4.2 vorgestellte Algorithmus unterteilt rekursiv die Linien an der Stelle ihrer maximalen Abweichung zur Konturkurve des Radarbildobjektes. Das heißt, er findet die Extrempunkte der Kurve bezüglich der approximierenden Linie. Diese Linien sind deshalb, was den mittleren Fehler der Annäherung zum Verlauf der Kontur anbelangt, nicht optimal. Eine Verschiebung der Linien, wie in Abbildung 4.11 (a), (b) dargestellt, könnte diesen Fehler verringern. Allerdings sind dann die Linien nicht mehr miteinander verbunden, so daß eine interne Darstellung der Uferlinien in Form von Polygonzügen nicht mehr möglich wäre. Eine Verschiebung einzelner Punkte des Polygonzuges würde die Beschreibung der Uferlinien mit Polygonzügen erhalten. Jedoch müßten in diesem Fall die mittleren Fehler der beiden angrenzenden Linien berücksichtigt werden. Das macht die Neuberechnung eines Punktes sehr aufwendig (siehe Beispiel in Abbildung 4.11 (c)). In beiden Fällen würden sich unerwünschte Nebeneffekte einstellen. Zum einen, daß in den Analyseergebnissen Punkte auftreten, die durch ihre Verschiebung keine realen Messungen mehr sind. Zum anderen können

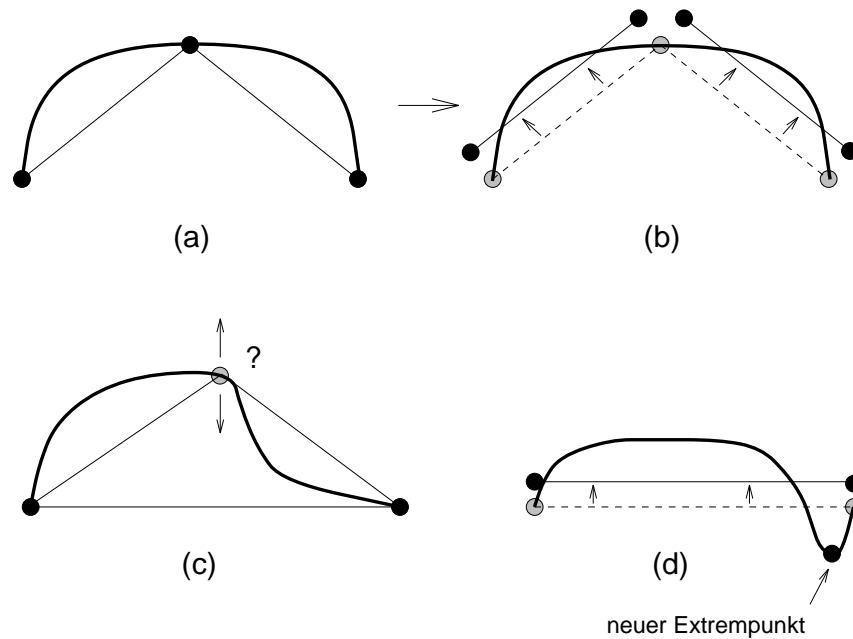


Abbildung 4.11: Verschiebung der approximierenden Linien

durch die Verschiebung der Linien neue Extrempunkte auftreten, die nicht erkannt werden. Ein Beispiel für diesen Fall ist in Abbildung 4.11 (d) zu sehen.

Der beschriebene Algorithmus ermöglicht einen zweiten Durchlauf über das Analyseergebnis, in dem für jede Linie überprüft wird, ob sie mit einer der beiden benachbarten Linien vereinigt werden kann. Einige Tests mit Uferkonturen haben jedoch gezeigt, daß das Ergebnis diesen zusätzlichen Rechenaufwand nicht rechtfertigt.

Es wäre denkbar, für die Approximation des Ufers außer den Linien noch weitere geometrische Primitive heranzuziehen. So könnte zum Beispiel die angemessene Beschreibung einer Kurve im Flußverlauf eine archimedische Spirale oder eine Ellipse sein. Eine interne Darstellung der Karte mit verschiedenen geometrischen Primitiven, wie zum Beispiel Linien und Ellipsen, wäre jedoch im Verlauf der Kartenerstellung wesentlich komplexer. Die Kombination der aktuellen Messungen aus dem Radarbild mit der Information aus den zurückliegenden Zeitschritten würde ebenfalls neue Probleme aufwerfen. Abbildung 4.12 zeigt ein Beispiel für eine Situation, in der es schwierig ist, die Ergebnisse der Approximation sinnvoll mit der bisher generierten Karte zu vereinigen. Die Entscheidung, ob zwei Objekte überhaupt miteinander verrechnet werden sollen, kann nur durch eine Berechnung des Abstandes der Objekte voneinander gefällt werden. Im Falle von Objekten mit nichtlinearen Gleichungen führt das zu aufwendigeren Rechnungen. In dieser Arbeit werden deshalb ausschließlich Linien zur Approximation der Objektkonturen verwendet.

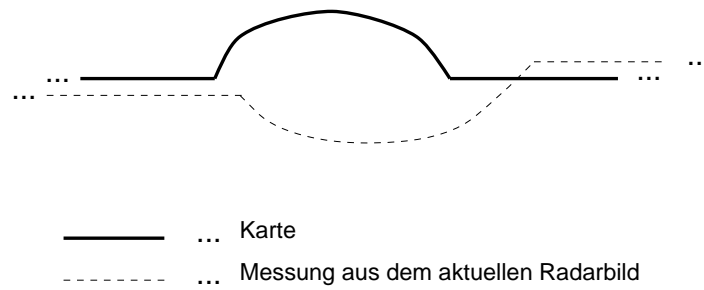


Abbildung 4.12: Approximation mit verschiedenen geometrischen Primitiven

Kapitel 5

Vereinigung der Analyseergebnisse aus aufeinanderfolgenden Zeitschritten

Nach der in Kapitel 4 beschriebenen Zerlegung des Radarbildes in kohärente Stücke und der Bestimmung der Eckpunkte ist die Analysearbeit getan. Die zusammenhängenden Stücke des Radarbildes sind in abstrakter und stark komprimierter Darstellung jeweils einem Polygonzug zugeordnet. Da sich die Analyseergebnisse aus aufeinanderfolgenden Zeitschritten überlappen, ist es für den Aufbau einer internen Karte erforderlich, die redundante Information wegzulassen und nur die relevante Information zu speichern. Deshalb müssen die Analyseergebnisse aus den einzelnen Zeitschritten miteinander kombiniert werden. In diesem Kapitel wird dieser Vorgang erläutert. Zunächst wird die interne Darstellung der Polygonzüge, also deren Datenstrukturen, vorgestellt. Es folgt eine Erklärung des Konzepts, mit dem die Positionen der Eckpunkte der Polygonzüge geschätzt werden: der *Kalman-Filter*. Anschließend wird dann der eigentliche Algorithmus zur Aktualisierung der Karte beschrieben.

5.1 Darstellung von Radarbild und Karte

5.1.1 Datenstrukturen

In Abbildung 5.1 sind noch einmal die einzelnen Stufen der Verarbeitung eines Radarbildes zusammen mit den zugehörigen Datenstrukturen aufgeführt.

Am Anfang der Verarbeitung liegt das Radarbild in der bereits in Abschnitt 3.2 vorgestellten *Run-Length-Codierung* vor. Der Bildinhalt ist darin als eine Menge von Echos

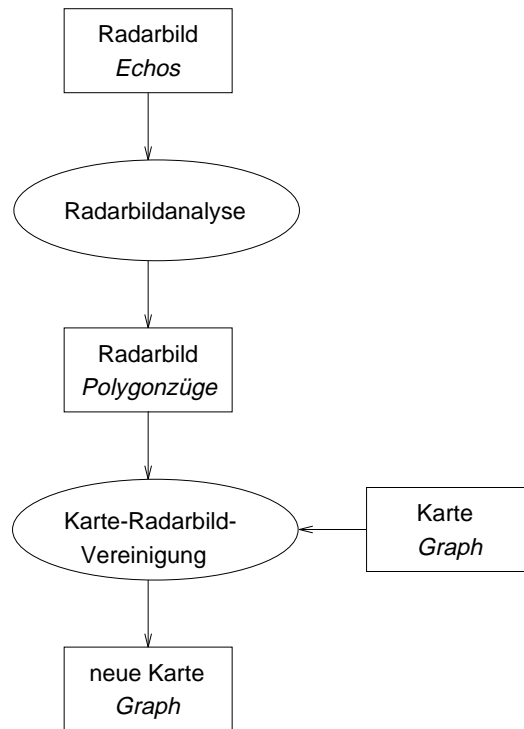


Abbildung 5.1: Datenstrukturen

beschrieben, die in einem zweidimensionalen Array abgelegt sind. Nach der Radarbildanalyse (Kapitel 4) liegt die relevante Information in Form einer Menge von Polygonzügen vor, die intern als verkettete Listen gespeichert sind (siehe Abbildung 5.2).

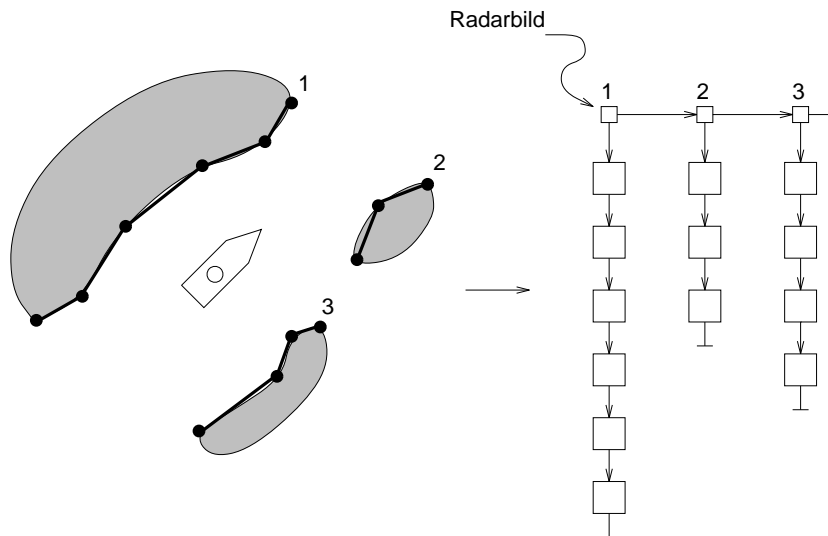


Abbildung 5.2: Analyseergebnis als Menge von Polygonzügen

Bei der Kartenerstellung im Laufe der Flußfahrt kommen zur bisher bestehenden Karte neue Punkte oder ganze Polygonzüge hinzu. Neue Punkte werden an die Punkte der Karte angehängt oder zwischen zwei Punkten der Karte eingefügt.

Grundsätzlich würde für die Beschreibung einer Uferlinie ein einfacher Polygonzug ausreichen. Es gibt jedoch Fälle, in denen derselbe Flußabschnitt zu verschiedenen Zeitpunkten im Radarbild unterschiedlich aussieht. In Abbildung 5.3 ist ein Beispiel für einen solchen Fall zu sehen. Im oberen Teil der Abbildung liegt das „Loch“ im Ufer noch im Hintergrund und wird deshalb bei der Analyse des Radarbildes nicht berücksichtigt. Im nächsten Zeitschritt kann jedoch in der Vorderkontur des Objekts eine Lücke auftreten, so daß eine Bucht durch die Radarbildanalyse erkannt wird (siehe Abbildung 5.3 unten). Solche Situationen treten bei Radarbildern relativ häufig auf. Bei der ausschließlichen Verwendung von Polygonzügen müßte dann sofort eine Entscheidung für die eine oder die andere Alternative getroffen werden. Es ist aber wünschenswert, eine solche Entscheidung noch hinauszuschieben, bis klar geworden ist, welche der beiden Alternativen dem tatsächlichen Uferverlauf am besten entspricht. Die Idee ist deshalb, *beide* Möglichkeiten in die Karte aufzunehmen und zu einem späteren Zeitpunkt die wahrscheinlichere auszuwählen. Eine Sequenz aus den beiden Radarbildern aus Abbildung 5.3 wird also so verarbeitet, daß die Karte beide potentiellen Uferverläufe repräsentiert (siehe Abbildung 5.4).

Dadurch können Punkte der Karte mehr als zwei Nachbarn besitzen. Die entstehende Datenstruktur der Karte ist also ein Graph.

5.1.2 Interne Darstellung

Dieser wachsende Graph bildet eine dynamische Datenstruktur, die sich durch statische Datentypen, wie zum Beispiel Felder, nicht mehr beschreiben läßt. Die Karte muß also durch zur Laufzeit erzeugte dynamische Variablen gespeichert werden. Dazu existiert ein Datentyp `Graphliste`, der die Uferabschnitte der Karte in einer doppelt verketteten Liste enthält. Die Uferabschnitte selbst sind durch Angabe eines Wurzelknotens definiert. Ein Knoten hat den Datentyp `KnotenTyp`, der alle relevanten Informationen über den entsprechenden Punkt in der Karte enthält. Dazu gehören

- Rechts- und Hochwert der Position im Gauß-Krüger-Koordinatensystem, im folgenden nur noch x - und y -Koordinaten genannt,
- ein Zeiger auf den Kalmanfilter des Knotens,
- verschiedene Markierungen, die für Suchvorgänge auf dem Graphen benötigt werden und
- eine Liste aller adjazenten Knoten.

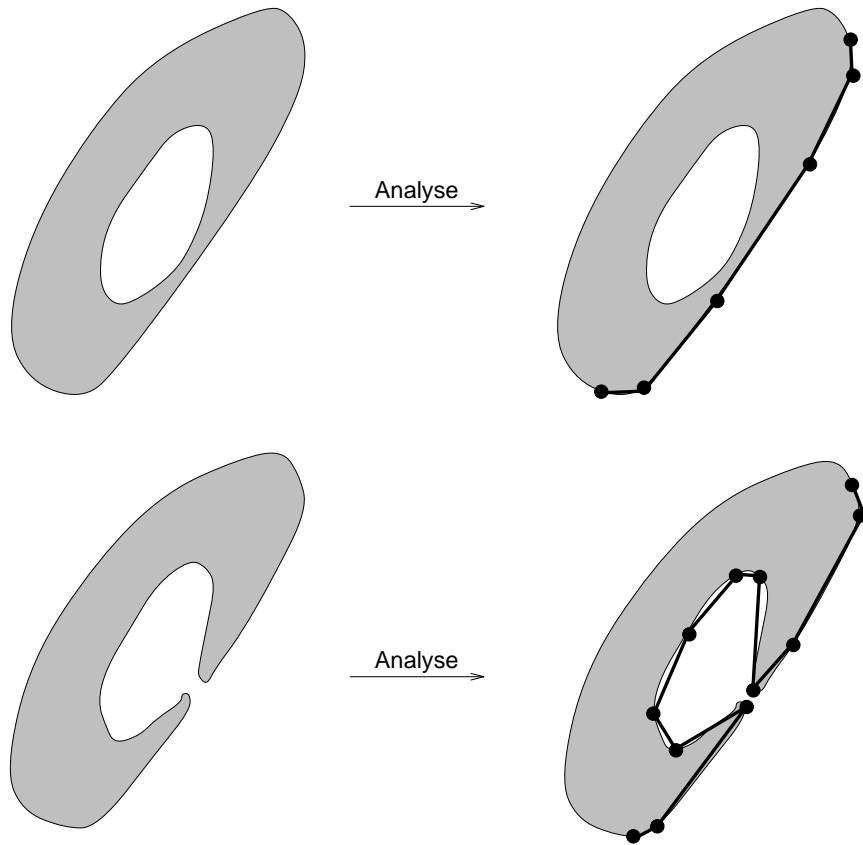


Abbildung 5.3: Varianten für den Uferverlauf

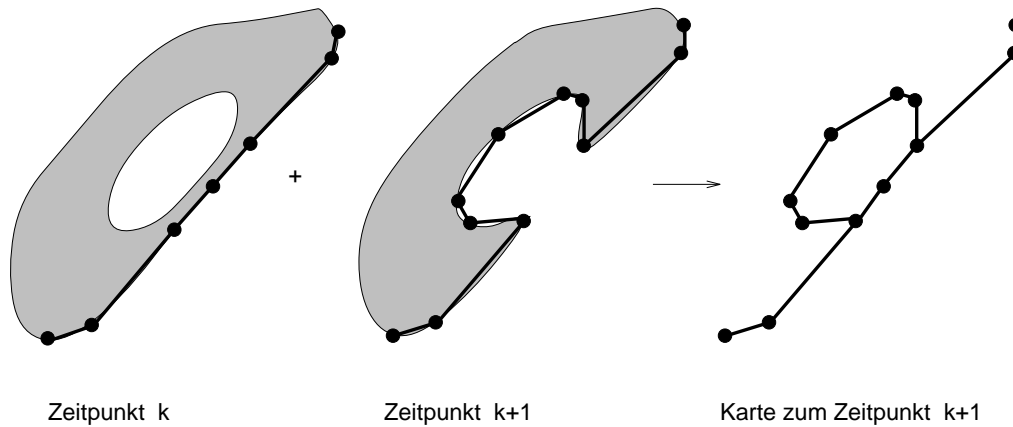


Abbildung 5.4: Karte mit Varianten für den Uferverlauf

Die daraus entstehende interne Darstellung der Karte ist in Abbildung 5.5 an einem Beispiel zu sehen.

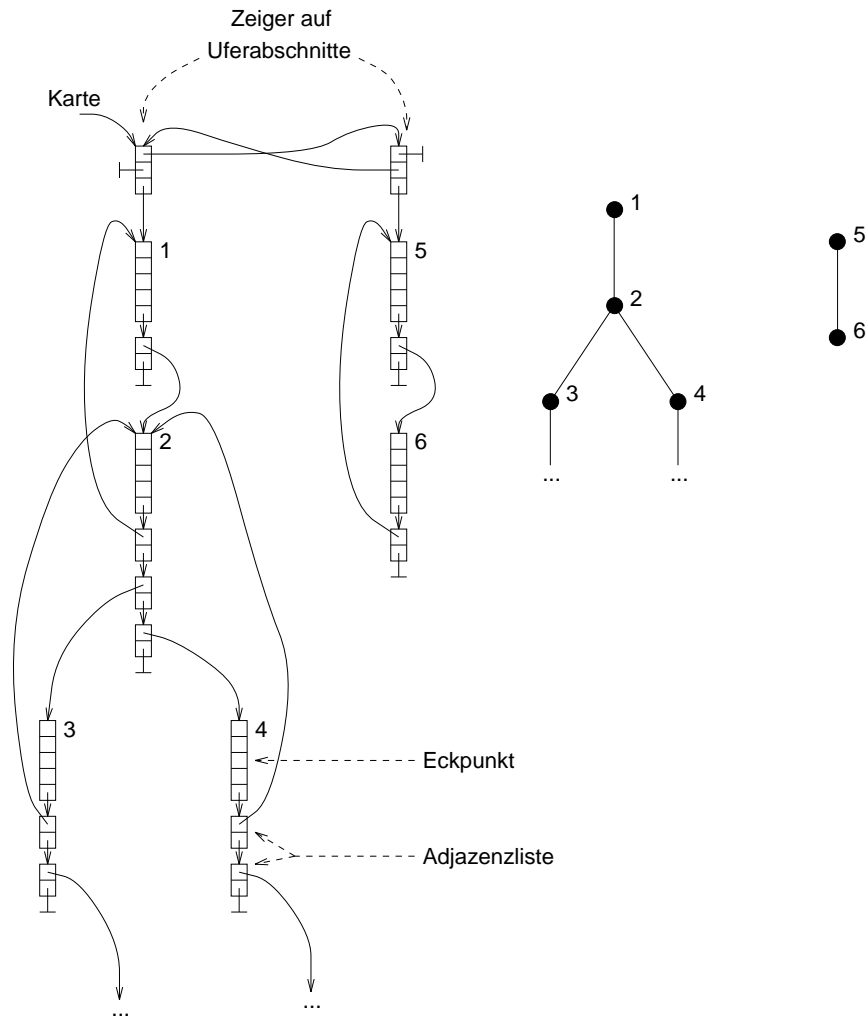


Abbildung 5.5: Interne Darstellung der Karte

5.2 Kalman-Filter

Ein Kalman-Filter ist ein Algorithmus, der bei bekannter Dynamik von System und Meßwerk sowie Informationen über das Rauschen, mit dem die Messungen behaftet sind, eine Schätzung für den Zustand des System bestimmt.

In Abbildung 5.6 ist der prinzipielle Aufbau des Kalman-Filters eines ungesteuerten Systems in Form eines Blockschaltbildes zu sehen. Die dabei verwendeten Bezeichnungen sind

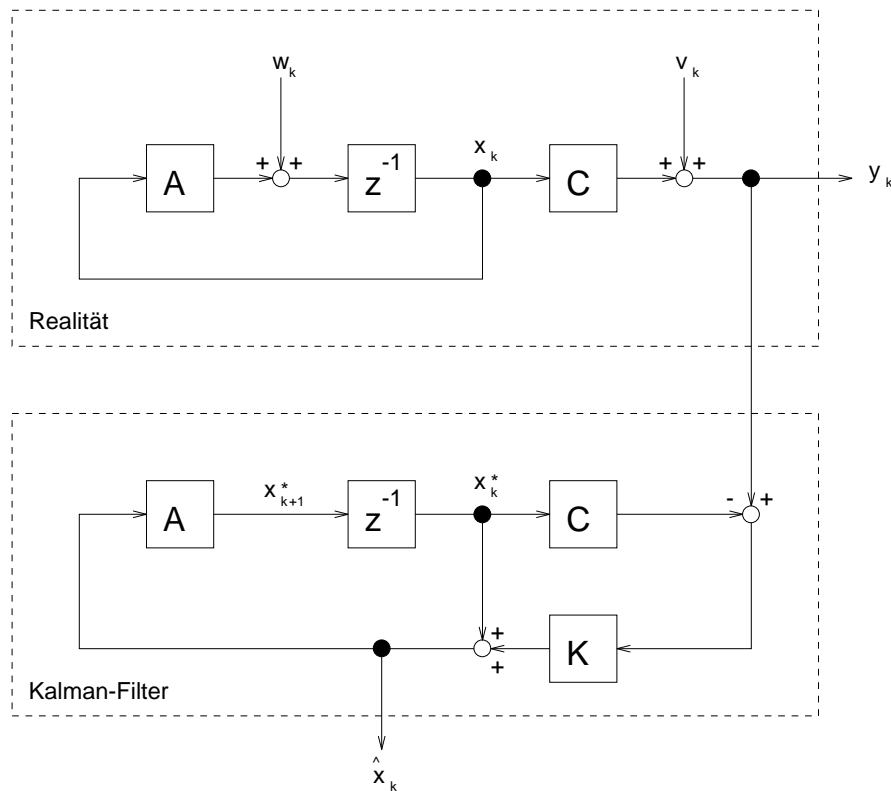


Abbildung 5.6: Aufbau eines Kalman-Filters

- \vec{x}_k ... Zustandsvektor zum Zeitpunkt k ,
- \vec{y}_k ... Meßvektor zum Zeitpunkt k ,
- A ... Systemmatrix,
- C ... Meßmatrix,
- \vec{w}_k ... Systemrauschen zum Zeitpunkt k , wird als weißes Rauschen¹ angenommen,
- \vec{v}_k ... Meßrauschen zum Zeitpunkt k , weißes Rauschen,
- K_k ... Kalman-Matrix zum Zeitpunkt k und
- z^{-1} ... Verzögerungsglied, das die Werte um einen Taktzyklus verzögert.

Im oberen Teil ist das reale System mit seinen Rauschprozessen \vec{w}_k und \vec{v}_k abgebildet. Die Ausgangsgröße ist der Meßvektor \vec{y}_k . Im unteren Teil ist das Modell zu sehen,

¹Ein weißes Rauschen ist ein stationäres, ergodisches, normalverteiltes Rauschsignal mit dem Mittelwert 0

das parallel zur Realität vom Kalman-Filter mitgerechnet wird. Dieses System liefert die Schätzung $\hat{\vec{x}}_k$ für den Zustand des realen Systems. Neben der Schätzung liefert der Kalman-Filter die Fehlerkovarianzmatrix \hat{P}_k , die eine Aussage über die Güte des Systems macht.

Die Kalman-Matrix K_k gewichtet den Prädiktionsfehler $\vec{y}_k - C\vec{x}_k^*$. Für ihre Berechnung werden noch folgende Werte benötigt:

- Q_k ... Kovarianzmatrix des Systemrauschens,
- R_k ... Kovarianzmatrix des Meßrauschens und
- P_k ... Fehlerkovarianzmatrix des Schätzvektors.

Dabei werden Q_k und R_k dem Algorithmus vorgegeben, während P_k intern berechnet wird. Die Matrix P muß nur einmal beim Start des Kalman-Filters initialisiert werden.

Der Algorithmus des Kalman-Filters läßt sich in eine Schätzphase und eine Anpassungsphase unterteilen. In der Schätzphase wird aus dem alten Zustandsvektor $\hat{\vec{x}}_k$ der geschätzte Zustandsvektor \vec{x}_{k+1}^* bestimmt und eine Schätzung für die Fehlerkovarianzmatrix P_{k+1}^* aufgestellt:

$$\vec{x}_{k+1}^* = A\hat{\vec{x}}_k \quad (5.1)$$

$$P_{k+1}^* = AP_kA^T + Q_k \quad (5.2)$$

Liegt die Messung \vec{y}_{k+1} vor, so beginnt die Anpassungsphase des Kalman-Filters. Hier werden mit Hilfe der neuen Werte in

$$K_{k+1} = P_{k+1}^*C^T(CP_{k+1}^*C^T + R_{k+1})^{-1} \quad (5.3)$$

der neue Zustandsvektor und seine Kovarianzen berechnet:

$$\hat{\vec{x}}_{k+1} = \vec{x}_{k+1}^* + K_{k+1}(\vec{y}_{k+1} - C_{k+1}\vec{x}_{k+1}^*) \quad (5.4)$$

$$\hat{P}_{k+1} = P_{k+1}^* - K_{k+1}C_{k+1}P_{k+1}^* \quad (5.5)$$

Hier ist der Zustand des Systems die Position des Polygonzugpunktes. Außerdem wird noch der Geschwindigkeitsvektor des Punktes mit in das Modell aufgenommen, um in einem späteren Schritt mit dieser Information bewegte Objekte, wie Schiffe, herausfiltern zu können. Der Zustandsvektor ist insgesamt also vierdimensional. Der zeitkontinuierliche Ansatz für die Systemgleichung lautet wegen $v_x = \dot{x}_{pos}$ und $v_y = \dot{y}_{pos}$

$$\dot{\vec{x}} = A\vec{x} \quad (5.6)$$

$$\begin{pmatrix} \dot{x}_{pos} \\ \dot{y}_{pos} \\ \dot{v}_x \\ \dot{v}_y \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_{pos} \\ y_{pos} \\ v_x \\ v_y \end{pmatrix}. \quad (5.7)$$

Diese Gleichung muß diskretisiert werden, da die Messungen nicht kontinuierlich, sondern zeitdiskret vorliegen. Die zeitdiskrete Systemgleichung lautet

$$\begin{pmatrix} x_{pos} \\ y_{pos} \\ v_x \\ v_y \end{pmatrix}_{k+1} = \begin{pmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{pos} \\ y_{pos} \\ v_x \\ v_y \end{pmatrix}_k. \quad (5.8)$$

T ist darin der zeitliche Abstand zweier Messungen, also die Umlaufzeit der Radaran-
tenne von ca. 2,3 Sekunden.

Gemessen wird nur die Position eines Punktes. Die Meßgleichung lautet deshalb

$$\vec{y} = C\vec{x} \quad (5.9)$$

$$\begin{pmatrix} x_{pos_{gemessen}} \\ y_{pos_{gemessen}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_{pos} \\ y_{pos} \\ v_x \\ v_y \end{pmatrix}. \quad (5.10)$$

Als Startwerte für die Matrix P werden gewählt:

$$P_1 = \begin{pmatrix} (10m)^2 & 0 & 0 & 0 \\ 0 & (10m)^2 & 0 & 0 \\ 0 & 0 & (0,1\frac{m}{s})^2 & 0 \\ 0 & 0 & 0 & (0,1\frac{m}{s})^2 \end{pmatrix}. \quad (5.11)$$

Die Matrix Q berechnet sich aus der Diagonalmatrix Q_{var} , die die Varianzen des kontinuierlichen Systems enthält:

$$Q = \int_0^T A Q_{var} A^T dt. \quad (5.12)$$

Mit der Matrix

$$Q_{var} = \begin{pmatrix} \sigma_{x_{pos}}^2 & 0 & 0 & 0 \\ 0 & \sigma_{y_{pos}}^2 & 0 & 0 \\ 0 & 0 & \sigma_{v_x}^2 & 0 \\ 0 & 0 & 0 & \sigma_{v_y}^2 \end{pmatrix} \quad (5.13)$$

ergibt sich daraus

$$Q = \begin{pmatrix} \sigma_{x_{pos}}^2 T + \frac{1}{3} \sigma_{v_x}^2 T^3 & 0 & \frac{1}{2} \sigma_{v_x}^2 T^2 & 0 \\ 0 & \sigma_{y_{pos}}^2 T + \frac{1}{3} \sigma_{v_y}^2 T^3 & 0 & \frac{1}{2} \sigma_{v_y}^2 T^2 \\ \frac{1}{2} \sigma_{v_x}^2 T^2 & 0 & \sigma_{v_x}^2 T & 0 \\ 0 & \frac{1}{2} \sigma_{v_y}^2 T^2 & 0 & \sigma_{v_y}^2 T \end{pmatrix}. \quad (5.14)$$

Die Elemente der Matrix Q_{var} werden konstant mit

$$\sigma_{x_{pos}} = 1m \quad (5.15)$$

$$\sigma_{y_{pos}} = 1m \quad (5.16)$$

$$\sigma_{v_x} = 0,1 \frac{m}{s} \quad (5.17)$$

$$\sigma_{v_y} = 0,1 \frac{m}{s} \quad (5.18)$$

vorgegeben. Die Kovarianzmatrix des Meßrauschens R_k muß in Abhängigkeit von der Güte der Messung gesetzt werden. Diese wird zum einen durch die Meßgenauigkeit des Radargerätes und zum anderen durch die Größe des Strukturelements, das in Abschnitt 3.3 beschrieben wurde, bestimmt. Um die Größe des Strukturelements zu berücksichtigen, müssen zunächst Δw und Δr (vergleiche Abbildung 3.4) in das kartesische Koordinatensystem umgerechnet werden. Eine brauchbare Vereinfachung stellt die Initialisierung

$$R = R_k = \begin{pmatrix} (1RUnit)^2 & 0 \\ 0 & (1RUnit)^2 \end{pmatrix} \quad (5.19)$$

dar. Die Initialisierung der Matrizen P , Q und R erfordert im allgemeinen viel Erfahrung und ausgiebige Tests. Deshalb erheben die hier angegebenen Werte natürlich nicht den Anspruch, optimal zu sein.

5.3 Aktualisierung der Karte

Nachdem mit den Datenstrukturen und dem Kalman-Filter die Mittel zur Darstellung der Karte beschrieben wurden, soll nun der Algorithmus, der die Radarbilder aus aufeinanderfolgenden Zeitschritten miteinander kombiniert, vorgestellt werden. Nach der Aufnahme jedes Radarbild findet folgender Ablauf statt:

FÜR ALLE Radarbild-Polygonzüge
Radarbild-Polygonzug mit der Karte kombinieren
Schätzphase der Kalman-Filters ausführen
Karte überarbeiten

Wie ein Radarbild-Polygonzug mit der Karte kombiniert wird und welche Sonderfälle dabei auftreten können, wird im folgenden Abschnitt besprochen. Anschließend wird für alle Kartenpunkte die Schätzphase ihres Kalman-Filters aktiviert (siehe Gleichungen 5.1 und 5.2). Zum Überarbeiten der Karte gehört, daß Punkte, die sehr nahe zusammenliegen vereinigt und solche Punkte, die sehr unwahrscheinlich geworden sind, eliminiert werden.

5.3.1 Radarbild-Polygonzug mit der Karte kombinieren

Um einen Radarbild-Polygonzug, im folgenden nur noch kurz *Polygonzug* genannt, der Karte zuzuordnen, genügt es im allgemeinen nicht, nur die Punkte von Polygonzug und Karte zu vergleichen. Es muß vielmehr nach Überlappungen zwischen dem Polygonzug und der Karte gesucht werden. Dabei sind folgende Fälle zu unterscheiden:

1. Der Punkt des Polygonzuges fällt mit einem Kartenpunkt zusammen.
2. Der Punkt des Polygonzuges liegt auf der Verbindungsstrecke zweier adjazenter Kartenpunkte.
3. Der Punkt des Polygonzuges kann nicht mit der Karte in Verbindung gebracht werden.

Im ersten Fall sollen die Koordinaten des Polygonzugpunkts mit denen des Kartenpunktes verrechnet werden, so daß für den Kartenpunkt eine neue, genauere Positionsbestimmung vorliegt. Im zweiten Fall wird der Punkt des Polygonzuges in die Verbindungsstrecke der zwei Kartenpunkte eingefügt. Im letzten Fall wird der Punkt als neuer Kartenpunkt initialisiert.

Bei der Vereinigung eines Polygonzuges mit der Karte wird der Polygonzug Punkt für Punkt betrachtet. Zunächst wird versucht, den ersten Punkt der Karte zuzuordnen. Gelingt dies, so wird solange versucht, den Polygonzug mit der Karte zur Deckung zu bringen, bis ein Punkt des Polygonzuges außerhalb der Karte liegt oder der Polygonzug zu Ende ist.

Kann der Punkt des Polygonzuges nicht der Karte zugeordnet werden, so wird er neu in der Karte initialisiert. Bevor jetzt aber der nächste Polygonzugpunkt betrachtet werden darf, muß die Karte nach Punkten durchsucht werden, die auf der Strecke zwischen dem letzten und dem nächsten Polygonzugpunkt liegen. Wird kein solcher Kartenpunkt gefunden, so wird wie beim ersten Punkt versucht, den nächsten Polygonzugpunkt der Karte zuzuordnen.

Liegt ein Kartenpunkt auf der Verbindungsstrecke der beiden Polygonzugpunkte, so wird wieder ausgehend von diesem Kartenpunkt versucht, den Polygonzug mit der Karte zur Deckung zu bringen.

Dieser Ablauf läßt sich mit dem endlichen Automat aus Abbildung 5.7 beschreiben.

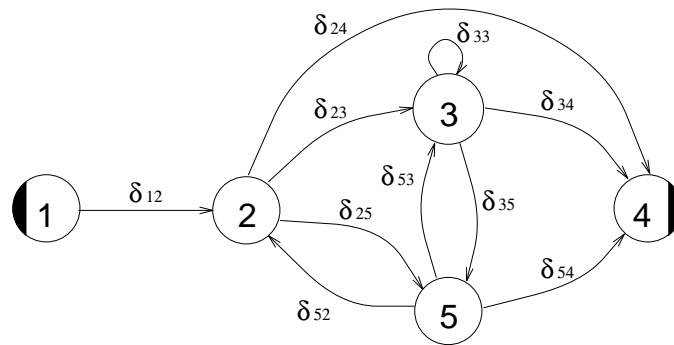


Abbildung 5.7: Beschreibung der Vereinigung von Polygonzug und Karte mit einem endlichen Automaten

Die Zustände des Automaten sind:

- 1 ... „Startzustand“,
- 2 ... „Polygonzugpunkt der Karte zuordnen“,
- 3 ... „Karte mit Polygonzug überdecken“,
- 4 ... „Endzustand“ und
- 5 ... „Kartenpunkt der Polygonzugstrecke zuordnen“.

Die Zustandsübergänge haben folgende Bedeutungen:

- δ_{12} ... „ersten Polygonzugpunkt nehmen“,
- δ_{23} ... „Polygonzugpunkt konnte zugeordnet werden“,
- δ_{25} ... „Polygonzugpunkt konnte nicht zugeordnet werden“,
- δ_{33} ... „Polygonzugpunkt innerhalb der Karte“,
- δ_{35} ... „Polygonzugpunkt außerhalb der Karte“,
- δ_{52} ... „kein Kartenpunkt auf der Polygonzugstrecke“,
- δ_{53} ... „Kartenpunkt auf der Polygonzugstrecke gefunden“ und
- $\delta_{24}, \delta_{34}, \delta_{54}$... „Polygonzug zu Ende“.

Im folgenden werden die einzelnen Zustände des Automaten genauer beschrieben.

Zustand 2: Polygonzugpunkt der Karte zuordnen

Im Zustand 2 des Automaten wird versucht, den nächsten Punkt P_P des Polygonzuges der Karte zuzuordnen. Dabei muß unterschieden werden, ob der Polygonzugpunkt einem Kartenpunkt P_K oder der Verbindungsstrecke $\overline{P_{K1}P_{K2}}$ zweier Kartenpunkte zuzuordnen ist. Wird P_P dem Kartenpunkt P_K zugeordnet, so wird P_P dem Kalman-Filter von P_K als Meßwert übergeben, so daß für P_K eine Neuberechnung der Position stattfindet. Liegt P_P in einem gewissen Bereich um $\overline{P_{K1}P_{K2}}$, so wird er in die Kartenstrecke als neuer Punkt eingefügt. In beiden Fällen muß außerdem entschieden werden, ob P_P der erste Punkt des Polygonzugs ist oder ob er einen Vorgänger hat. Im letzteren Fall wird der Vorgänger von P_P an die Karte angehängt. In Abbildung 5.8 sind diese Fälle an einem Beispiel zu sehen.

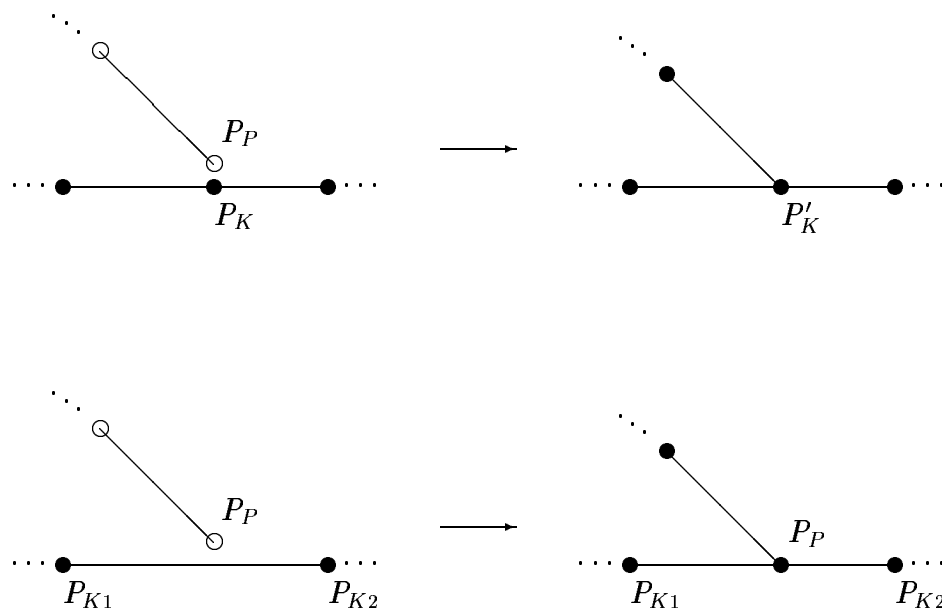


Abbildung 5.8: Zuordnung eines Polygonzugpunktes zur Karte

Oben liegt P_P in einem gewissen Bereich um P_K , im folgenden *Einzugsbereich* von P_K oder $K(P_K)$ genannt. P_K erhält deshalb P_P als Meßwert für die eigene Position. Der Vorgänger von P_P wird als Variante an P_K angehängt. Im unteren Teil der Abbildung liegt P_P im Einzugsbereich der Strecke $\overline{P_{K1}P_{K2}}$, im folgenden geschrieben als $B(\overline{P_{K1}P_{K2}})$. P_P wird deshalb in die Kartenstrecke eingefügt und der Vorgänger von P_P , falls dieser existiert, als Variante angehängt. Kann P_P weder einem Kartenpunkt, noch einer Kartenstrecke zugeordnet werden, so wird P_P in die Karte als neuer Punkt aufgenommen. Der Automat wechselt in diesem Fall nach Zustand 5. Daraus ergibt sich der in Abbildung 5.9 dargestellte Ablauf für den Zustand 2 des Automaten.

Versuche, Polygonzugpunkt P_P der Karte zuzuordnen		
FALLS	$P_P \in K(P_K)$	
DANN	P_K Meßwert P_P zuweisen	
	Vorgänger von P_P als Variante anhängen	
	FALLS Polygonzug zu Ende	
	DANN Übergang in Zustand 4	
	SONST Übergang in Zustand 3	
SONST	FALLS $P_P \in B(\overline{P_{K_1}P_{K_2}})$	
	DANN	P_P in Karte einfügen
		Vorgänger von P_P als Variante anhängen
		FALLS Polygonzug zu Ende
		DANN Übergang in Zustand 4
		SONST Übergang in Zustand 3
	SONST	FALLS Polygonzug zu Ende
		DANN Übergang in Zustand 4
		SONST Übergang in Zustand 5

Abbildung 5.9: Struktogramm für Zustand 2

Um den Punkt P_P der Karte zuzuordnen, muß die Karte durchsucht werden. Für die Entscheidung, ob $P_P \in K(P_K)$ muß jeder Knoten und für $P_P \in B(\overline{P_{K_1}P_{K_2}})$ jede Kante des Graphen betrachtet werden. Das wird erreicht durch eine rekursive Tiefensuche in dem Graphen der Karte. Jeder Knoten P , der besucht wird, erhält zunächst eine Markierung, um zu verhindern, daß der Knoten ein zweites Mal besucht wird. Dann werden nacheinander alle Nachbarknoten P_i von P betrachtet und überprüft, ob P_P im Einzugsbereich von P_i oder auf der Strecke $\overline{PP_i}$ liegt. Wenn ja, so wird die Tiefensuche abgebrochen und P bzw. P und P_i zurückgegeben.

Für den Test ob $P_P \in K(P_i)$ ist, werden zunächst die Zustandskovarianzen aus der P-Matrix des Kalmanfilters von P_i ausgelesen. Der Radius des Einzugsbereiches wird dann aus

$$r = 3\sqrt{\frac{\sigma_{x_{\text{pos}}}^2 + \sigma_{y_{\text{pos}}}^2}{2}} \quad (5.20)$$

berechnet. Das heißt, der Einzugsbereich von P_i wird durch einen Kreis mit dem Radius der dreifachen Standardabweichung σ der Position angenähert. Da es sich bei den Rauschsignalen um weiße Rauschen handelt (siehe Abschnitt 5.2), ist die Dichtefunktion der Meßfehler eine Gaußsche Glockenkurve. In den Bereich $\pm 3\sigma$ um den Mittelwert fallen dann statistisch gesehen 99,7 % aller Meßwerte für den Punkt. Die Entscheidung, ob der Punkt P_P in $K(P_i)$ enthalten ist, wird jetzt gefällt durch die

Abfrage

$$\boxed{P_P \in K(P_i) \iff |\overline{P_P P_i}| \leq r.} \quad (5.21)$$

Um zu entscheiden, ob $P_P \in B(\overline{P_{K_1} P_{K_2}})$, wird ein rechteckiger Bereich um die Strecke $\overline{P_{K_1} P_{K_2}}$ festgelegt. Der Abstand b der Rechteckseite zur Strecke $\overline{P_{K_1} P_{K_2}}$, also die halbe Rechteckbreite, wird berechnet als

$$b = mf. \quad (5.22)$$

Dabei ist m der Parameter für den Approximationsalgorithmus, der die Schranke bestimmt, ab der eine Linie nicht mehr weiter unterteilt wird (siehe dazu Abschnitt 4.2). f ist ein konstanter Faktor, der den erwarteten Fehler für die Positionen der Punkte P_{K_1} , P_{K_2} und P_P berücksichtigt. Dieser ist mit

$$f = 1,5 \quad (5.23)$$

voreingestellt. In Abbildung 5.10 ist der Einzugsbereich um eine Strecke $\overline{P_{K_1} P_{K_2}}$ dargestellt.

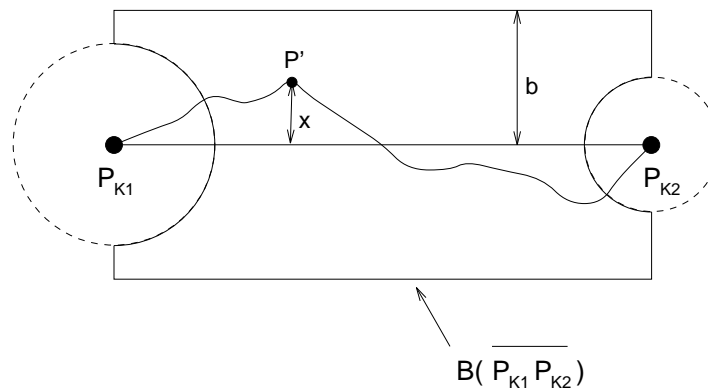


Abbildung 5.10: Einzugsbereiche

Die Aussparungen um die Punkte P_{K_1} und P_{K_2} entstehen, da zunächst deren Einzugsbereiche auf Enthaltensein von P_P geprüft werden. An dem angedeuteten Verlauf der Kontur, die durch $\overline{P_{K_1} P_{K_2}}$ approximiert wurde, ist auch ersichtlich, warum $b \geq m$ sein muß. Denn nur dann kann ein Punkt P' , dessen Abstand zur Strecke x fast m war und im nächsten Schritt als Eckpunkt in einem Polygonzug erkannt wird, der Strecke $\overline{P_{K_1} P_{K_2}}$ zugeordnet werden.

Um nun zu entscheiden, ob $P_P = (x_p, y_p)$ in dem Bereich zwischen den Punkten $P_{K_1} = (x_1, y_1)$ und $P_{K_2} = (x_2, y_2)$ enthalten ist, wird zunächst das Koordinatensystem O in den Punkt P_{K_1} verschoben. In diesem Koordinatensystem A haben die Punkte

die Koordinaten $P_P = (x_p - x_1, y_p - y_1)$, $P_{K1} = (0, 0)$ und $P_{K2} = (x_2 - x_1, y_2 - y_1)$. Anschließend wird die x -Achse in Richtung der Strecke $\overline{P_{K1}P_{K2}}$ gedreht. Im resultierenden Koordinatensystem B ergeben sich wegen

$$\Theta = \arctan \frac{y_2 - y_1}{x_2 - x_1} \tag{5.24}$$

$${}^A_B T = \begin{pmatrix} \cos \Theta & -\sin \Theta \\ \sin \Theta & \cos \Theta \end{pmatrix} \tag{5.25}$$

die Koordinaten von P_P zu

$${}^B P_P = \begin{pmatrix} {}^B P_{P_x} \\ {}^B P_{P_y} \end{pmatrix} = {}^A_B T^{-1} \cdot {}^A P_P = \begin{pmatrix} \cos \Theta & \sin \Theta \\ -\sin \Theta & \cos \Theta \end{pmatrix} \begin{pmatrix} x_p - x_1 \\ y_p - y_1 \end{pmatrix}. \tag{5.26}$$

In Abbildung 5.11 sind die Koordinatensysteme dieser Transformationen zu sehen.

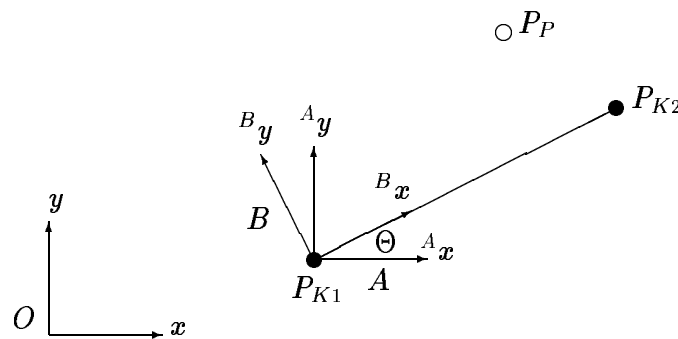


Abbildung 5.11: Koordinatensysteme beim Test auf Enthaltensein eines Punktes in einer Strecke

Jetzt kann mit einer einfachen Abfrage entschieden werden, ob P_P im Einzugsbereich der Strecke $\overline{P_{K1}P_{K2}}$ liegt:

$$P_P \in B(\overline{P_{K1}P_{K2}}) \iff 0 \leq {}^B P_{P_x} \leq {}^B P_{K2_x} \quad \wedge \quad |{}^B P_{P_y}| \leq b.$$

Zustand 3: Karte mit Polygonzug überdecken

Nachdem ein Polygonzugpunkt der Karte zugeordnet werden konnte, sollen jetzt im Zustand 3 des Automaten die nächsten Kanten des Polygonzuges mit den Kanten der Karte überdeckt werden. Es wird also versucht, den Pfad im Graphen der Karte zu finden, der der Beobachtung aus dem aktuellen Radarbild entspricht. Ausgangspunkt dieser Suche ist der letzte, der Karte zugeordnete Polygonzugpunkt P und der nächste Punkt P' des Polygonzuges (siehe Abbildung 5.12).

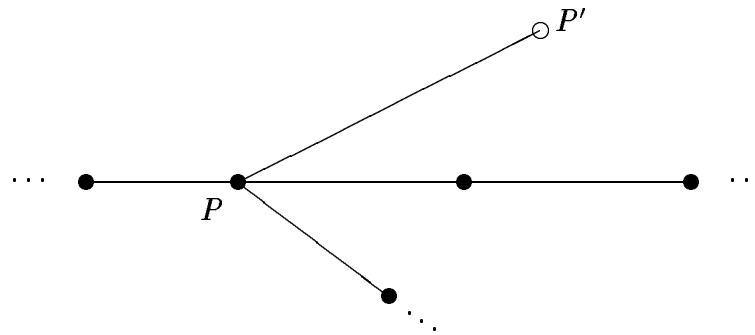


Abbildung 5.12: Ausgangssituation im Zustand 3

Um jetzt die Strecke $\overline{PP'}$ in der Karte wiederzufinden, werden alle Nachbarn P_i des Kartenpunktes P betrachtet und nacheinander die Fälle

1. $P' \in K(P_i)$
2. $P_i \in B(\overline{PP'})$
3. $P' \in B(\overline{PP_i})$

untersucht.

Im ersten Fall, $P' \in K(P_i)$, liegt der nächste Polygonzugpunkt P' im Einzugsbereich des Nachbarn P_i von P . Es wurde also die Kartenstrecke $\overline{PP_i}$ mit der Polygonzugstrecke $\overline{PP'}$ überdeckt (siehe Abbildung 5.13).

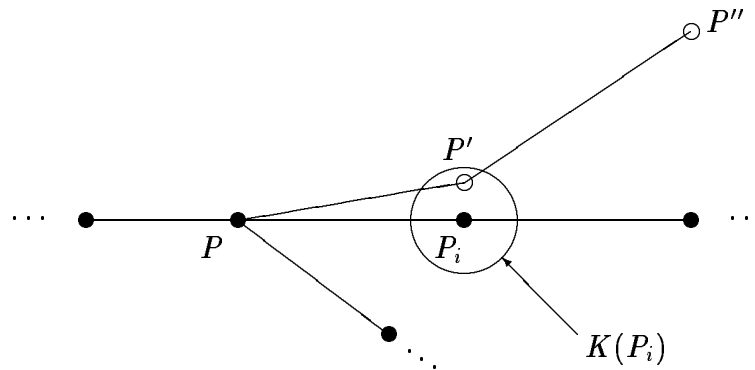


Abbildung 5.13: Zustand 3, erster Fall

Neuer Ausgangspunkt ist jetzt P_i mit dem nächsten Polygonzugpunkt P'' . Falls dieser nicht existiert, das heißt, wenn der Polygonzug zu Ende ist, findet ein Übergang in den Endzustand 4 statt. Ansonsten bleibt der Automat in Zustand 3.

Im zweiten Fall, $P_i \in B(\overline{PP'})$, wurde der Nachbar P_i von P gefunden, der im Einzugsbereich der Strecke zwischen P und P' liegt. Es wurde also ein Teil der Polygonzugstrecke $\overline{PP'}$ in der Karte erkannt. Dieser Fall ist in Abbildung 5.14 zu sehen.

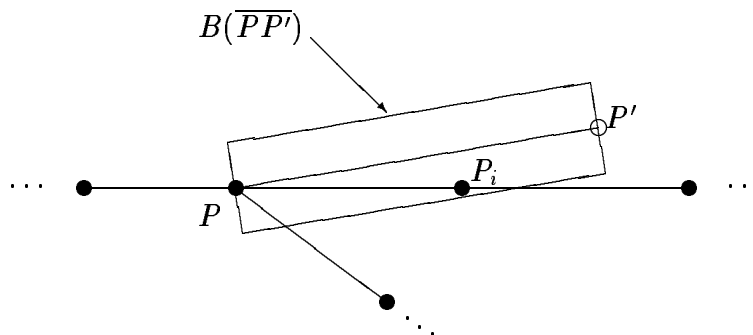


Abbildung 5.14: Zustand 3, zweiter Fall

Um den Rest der Strecke, also $\overline{P_iP'}$, auch noch der Karte zuzuordnen, bleibt der Automat in Zustand 3 mit den beiden Punkten P_i und P' .

Der letzte Fall $P' \in B(\overline{PP_i})$ bedeutet, daß die Strecke $\overline{PP'}$ mit einer Teilstrecke von $\overline{PP_i}$ assoziiert werden kann (siehe dazu Abbildung 5.15).

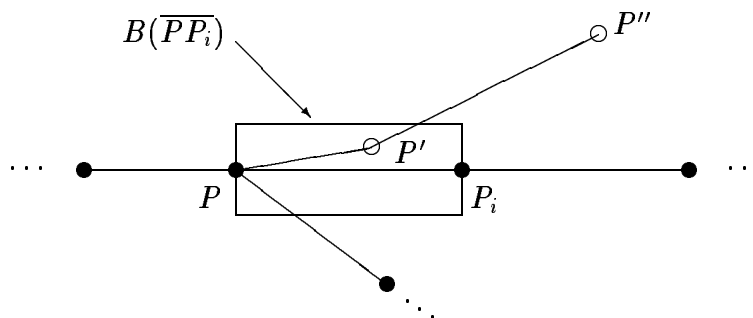


Abbildung 5.15: Zustand 3, dritter Fall

P' wird als neuer Kartenpunkt eingefügt und wird zusammen mit P'' neuer Ausgangspunkt für die Suche in Zustand 3. Wie im ersten Fall findet ein Übergang in den Endzustand 4 statt, falls P'' nicht existiert.

Wenn keiner der drei Fälle zutrifft, liegt die Strecke $\overline{PP'}$ außerhalb des betrachteten Graphen der Karte (siehe Abbildung 5.16).

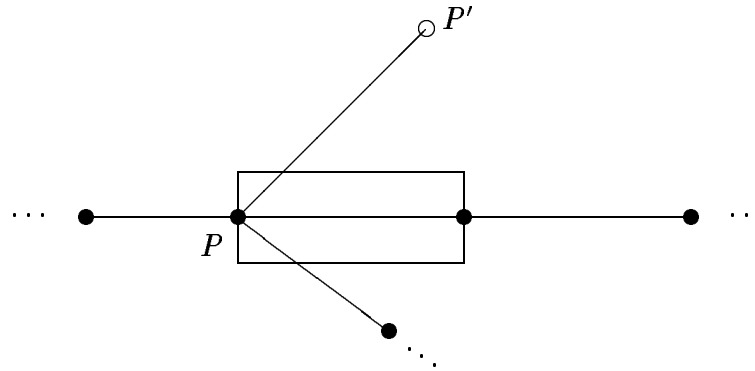


Abbildung 5.16: Zustand 3, Ende der Überdeckung

Diese Strecke stellt also eine neue Variante für diesen Graphen dar und wird deshalb an den Graphen angehängt und es findet ein Übergang in Zustand 5 des Automaten statt.

Der Zustand 3 enthält so, wie er bis jetzt beschrieben wurde, noch eine Unzulänglichkeit. Wie in Abbildung 5.17 zu sehen, ist es möglich, daß

$$P_i \in B(\overline{PP'}) \quad \text{und} \quad (5.27)$$

$$P \in B(\overline{P_iP'}) \quad (5.28)$$

gleichzeitig erfüllt sind.

Das kann zu einer Endlosschleife führen, bei der der Ausgangspunkt ständig zwischen P und P_i wechselt. Dieser Fall läßt ausschließen, in dem vor der Fallunterscheidung nur solche Nachbarknoten P_i von P betrachtet werden, die näher an P' liegen, als P selbst. Für den Nachbarknoten P_i muß also gelten:

$$|\overline{P_iP'}| < |\overline{PP'}|. \quad (5.29)$$

Durch diese zusätzliche Bedingung wird garantiert, daß sich der Ausgangspunkt für den nächsten Schritt, P , immer näher auf P' zubewegt.

nicht der Karte zugeordnet werden kann, die Karte nach Punkten durchsucht werden, die auf der Strecke zwischen dem letzten Ausgangspunkt, in diesem Beispiel P_3 , und dem nächsten Polygonzugpunkt P' liegen. Der Ablauf im Zustand 5 wird durch das Struktogramm in Abbildung 5.19 beschrieben. Dabei sind P der letzte Punkt, der der Karte zugeordnet werden konnte und P' der nächste Polygonzugpunkt.

Suche Kartenpunkte, die auf der Strecke $\overline{PP'}$ liegen	
Wähle den Punkt P_K aus, für den gilt: $ \overline{PP_K} \stackrel{!}{=} \min$	
FALLS	P_K existiert
DANN	P_K in die Strecke $\overline{PP'}$ einfügen
	P_K wird neuer Ausgangspunkt
	Übergang in Zustand 3
SONST	P' wird neuer Ausgangspunkt
	Übergang in Zustand 2

Abbildung 5.19: Struktogramm für Zustand 5

Im ersten Schritt wird die Karte mit einer rekursiven Tiefensuche nach Punkten durchsucht, die auf der Strecke $\overline{PP'}$ liegen. Dabei werden für jeden gefundenen Punkt P_K die folgenden Fälle unterschieden:

1. $P_K \in K(P)$, „Kartenpunkt P_K liegt im Einzugsbereich von P “,
2. $P_K \in B(\overline{PP'})$, „Kartenpunkt P_K liegt im Einzugsbereich der Strecke $\overline{PP'}$ “ oder
3. $P_K \in K(P')$, „Kartenpunkt P_K liegt im Einzugsbereich von P' “.

Anschließend wird der Punkt mit dem minimalen Abstand zu P gemäß der obigen drei Fälle mit der Strecke verrechnet (siehe Abbildung 5.8). Nachdem jetzt ein Kartenpunkt der Polygonzugstrecke zugeordnet werden konnte, wird jetzt im Zustand 3 des Automaten versucht, die Karte weiter mit dem Polygonzug zu überdecken. Im Beispiel in Abbildung 5.18 wird also zunächst P_4 als nächster Kartenpunkt gefunden und anschließend im Zustand 3 die Punkte P_5 und P_6 mit der Polygonzugstrecke kombiniert.

Wird in der Karte kein Punkt gefunden, der auf der Strecke $\overline{PP'}$ liegt, so findet ein Übergang nach Zustand 2 des Automaten statt. Dort wird eine Kartenstrecke gesucht, in deren Einzugsbereich P' liegt.

Ein Beispiel für die Arbeitsweise des Automaten mit seinen Zustandsübergängen ist in Abbildung 5.20 zu sehen. Die Punkte sind in der gleichen Weise bezeichnet wie in den vorangegangenen Beschreibungen der einzelnen Zustände.

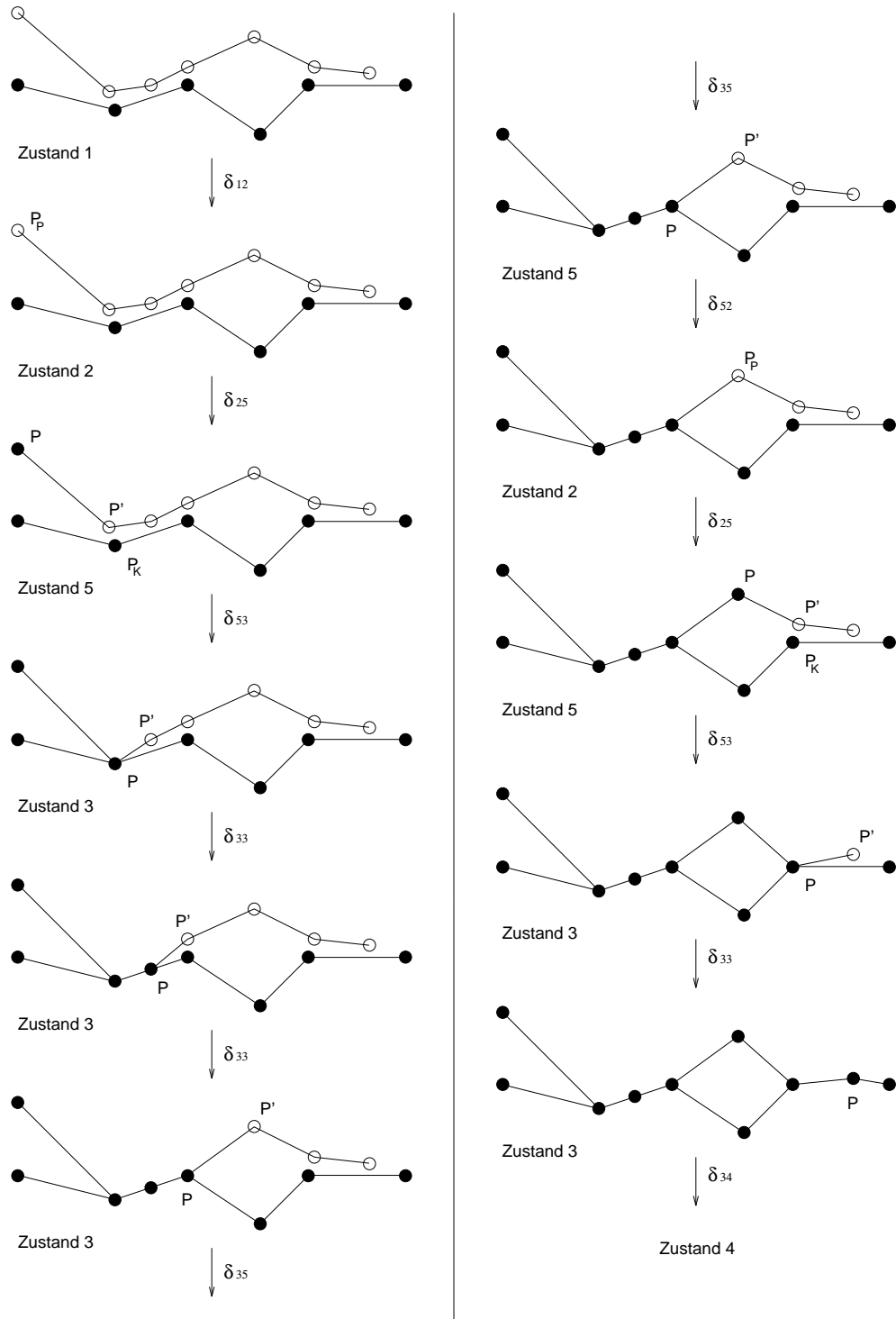


Abbildung 5.20: Arbeitsweise des Automaten

5.3.2 Überarbeiten der Karte

Nachdem die Polygonzüge aus dem letzten Radarbild durch die oben beschriebenen Abläufe mit der Karte kombiniert wurden, findet ein weiterer Verarbeitungsschritt auf der Karte statt. Durch die Aktualisierung der Positionen der Kartenpunkte mit den Meßpunkten aus den Polygonzügen ist es möglich, daß zwei Kartenpunkte sehr nahe zusammenrücken oder zusammenfallen. Um das zu erkennen und solche Punkte verschmelzen zu lassen, wird in einem weiteren Durchlauf über alle Punkte der Karte überprüft, ob Nachbarn eines Punktes in dessen Einzugsbereich liegen.

Dies geschieht wiederum mit einer rekursiven Tiefensuche in der gesamten Karte. An jedem erreichten Punkt P findet der in Abbildung 5.21 dargestellte Ablauf statt.

Markiere P	
FÜR ALLE Nachbarn P_i von P	
FALLS	$P_i \in K(P)$
DANN	verschmelze P und P_i
WIEDERHOLE, BIS keine Veränderung mehr am Knoten P	

Abbildung 5.21: Struktogramm: Vereinigen von Kartenpunkten

Wenn zwei Punkte P und P_i vereinigt werden sollen, so erhält derjenige Punkt mit der kleineren Zustandskovarianz, also der sicherere, den anderen Punkt als Meßwert. In die Messungskovarianzmatrix R werden dabei die Zustandskovarianzen aus der Matrix P des „Meßpunktes“ eingesetzt. Beim Vereinigen der Adjazenzlisten beider Knoten muß verhindert werden, daß gemeinsame Nachbarn beider Knoten zu doppelten Einträgen führen. In Abbildung 5.22 links liegen die beiden Nachbarn P_1 und P_2 im Einzugsbereich von P . Im mittleren Teil wurde P mit P_1 vereinigt. Rechts ist das Endergebnis nach der Vereinigung von P mit P_2 zu sehen. P hat die Nachbarn von P_1 und P_2 als neue Nachbarn bekommen. Eine Doppelkante zu P_3 wurde verhindert.

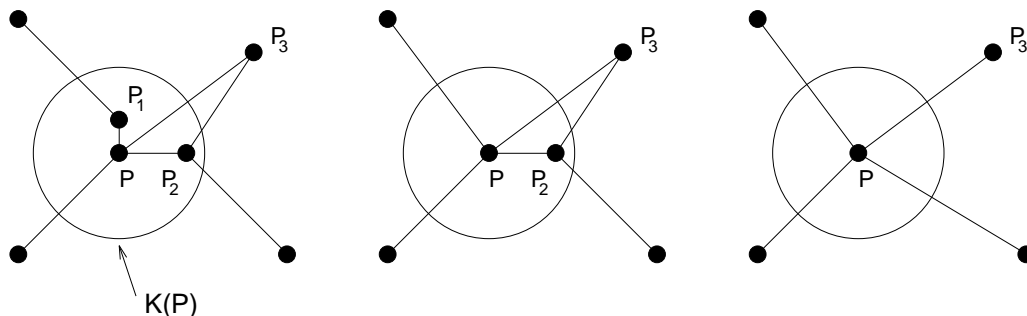


Abbildung 5.22: Vereinigen von Kartenpunkten

5.4 Beispiel

In den Abbildungen 5.23 und 5.24 sind Beispiele für generierte Karten nach einer Sequenz von ca. 100 Radarbildern zu sehen.

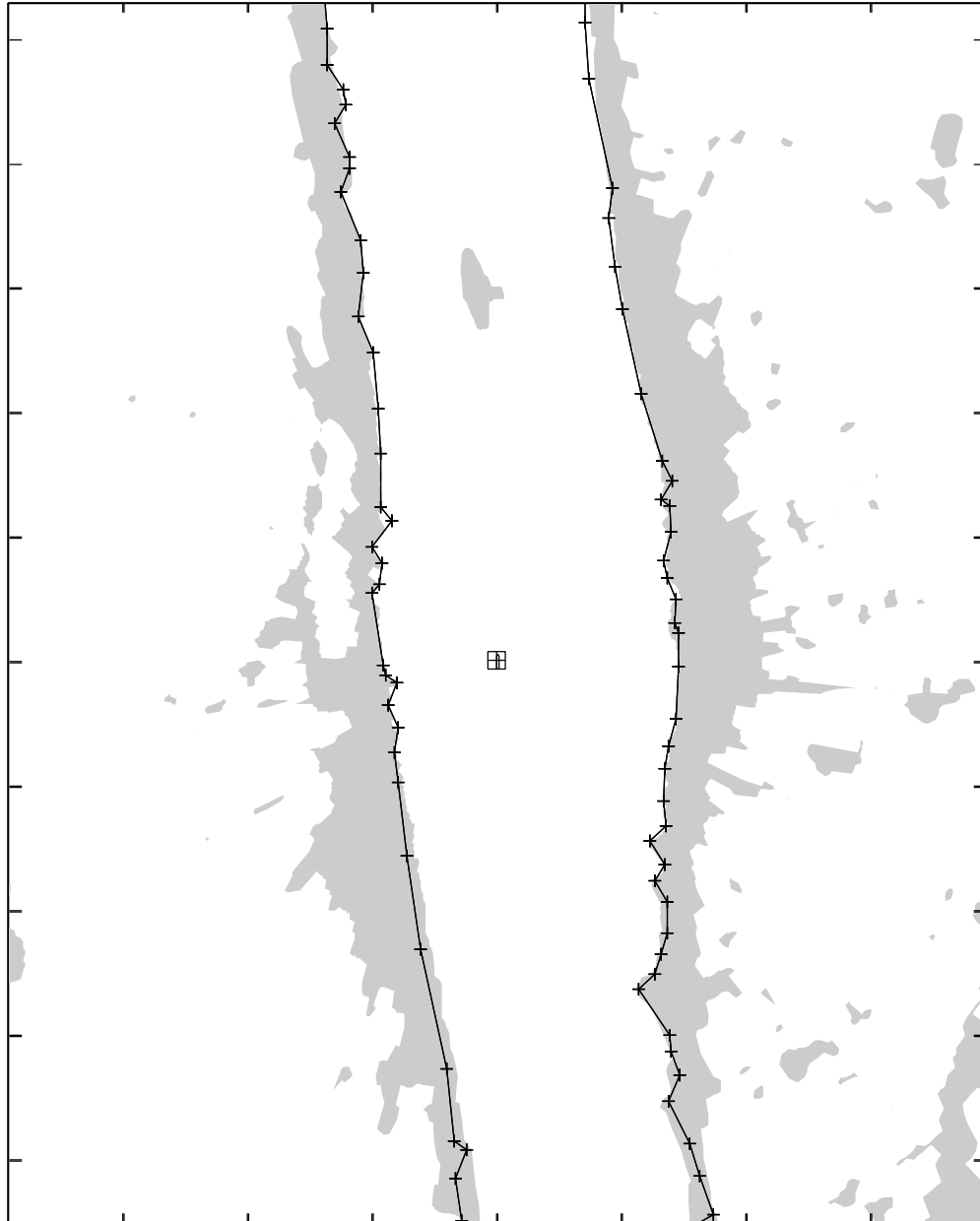


Abbildung 5.23: Beispiel 1: Kartenausschnitt des Rheins



Abbildung 5.24: Beispiel 2: Kartenausschnitt des Rheins

Kapitel 6

Bewertung und Ausblick

In dieser Arbeit wurde ein Verfahren zur automatischen Erstellung einer elektronischen Flußkarte vorgestellt. Wie viele Versuche gezeigt haben, lassen sich damit Radarbilder aus aufeinanderfolgenden Zeitschritten zu einer einzigen Darstellung der Uferabschnitte kombinieren. Es wurden Karten von verschiedenen Rhein- und Neckarabschnitten erzeugt. Vergleiche mit den bereits vorhandenen Karten haben gezeigt, daß die Ergebnisse präzise sind.

In einem ersten Schritt werden die Radarbilder, die durch Meßfehler des Radargerätes recht stark verrauscht sind, mit einem morphologischen Filter vorverarbeitet. Durch diese Vorfilterung werden Lücken in den Uferlinien geschlossen und die Konturen der Objekte geglättet. Es wurde gezeigt, daß der in Kapitel 4 beschriebene Approximationsalgorithmus dadurch die Objektkonturen durch weniger Linien darstellen kann, als bei einer Verarbeitung des verrauschten Originalbildes.

Die Approximation geschieht durch die rekursive Unterteilung einer approximierenden Linie an der Stelle ihrer maximalen Abweichung zur Kontur des Objektes. Es entsteht eine Darstellung des Objektes in Form eines Polygonzuges. Diese wird anschließend mit dem sogenannten *significance*-Kriterium überarbeitet (siehe Abbildung 4.7). Dieses Kriterium kann eventuell durch eine Entscheidung aufgrund des quadratischen Fehlers der einzelnen Repräsentationen noch verbessert werden.

Zur Vereinigung der Analyseergebnisse aus aufeinanderfolgenden Zeitschritten wurde ein Verfahren vorgestellt, daß auf einem endlichen Automaten basiert und Punkt für Punkt das Analyseergebnis mit der bisher bestehenden Karte kombiniert.

Es ist zu beobachten, daß viele Punkte in den nächsten Schritten nicht wiedererkannt werden. Der Grund dafür liegt in der Form der Uferabschnitte. Oft ist die Lage der Extrempunkte, die im Approximationsschritt gefunden werden, nicht eindeutig. Dadurch wird eine Kontur durch eine Vielzahl von Punkten dargestellt, die erst durch den in Abschnitt 5.3.2 beschriebenen Überarbeitungsschritt wieder verringert wird. Durch

diesen Wechsel zwischen Einfügen und Löschen von Kartenpunkten gehen Informationen über den eigentlichen Uferverlauf verloren. Ein anderer Ansatz wäre deshalb, äquidistante Stützstellen zu verwenden und die Uferkonturen zum Beispiel durch Splines darzustellen.

Die Laufzeit der Algorithmen wird durch die Suchoperationen auf der Karte in den Zuständen 2 und 5 dominiert. Wegen der dort eingesetzten Tiefensuche ist der Zeitaufwand insgesamt linear in der Anzahl der Kanten des Kartengraphen. Da der Graph hauptsächlich aus Polygonzügen besteht (siehe Abbildungen 5.23 und 5.24), kann ein linearer Zusammenhang zwischen der Anzahl der Kanten und der Anzahl der Knoten des Graphen angenommen werden. Dadurch sind die Algorithmen für kleine Karten echtzeitfähig.

Es wäre möglich, die Karte in gleichbleibender Größe zu halten, indem laufend solche Kartenpunkte, die nicht mehr durch Messungen aus dem aktuellen Radarbild bestätigt werden können, aus der internen Darstellung entfernt und zum Beispiel in eine Datei geschrieben werden.

Um die automatisch erzeugten Flußkarten im Schiffsprojekt verwenden zu können, müssen sie noch in die Darstellung der bisher vorhandenen Flußkarten überführt werden.

Anhang A

Befehle für den Kommandointerpreter

Um die Einstellung verschiedener Parameter der Module zur Laufzeit zu ermöglichen, können die einzelnen Module dem Kommandointerpreter Befehle zur Verfügung stellen. Im folgenden werden die Kommandos für die automatische Kartengenerierung angegeben.

A.1 Das Kommando `chartcreate`

Einstellbare Parameter

<code>Kartengenerierung einschalten?</code>	Ein- und Ausschalten der Kartengenerierung
---	--

A.2 Das Kommando `chartdraw`

Einstellbare Parameter

<code>Radarbildapproximation zeichnen?</code>	Zeichnet die Linien der letzten Approximation in das Radarbild ein
<code>Karte zeichnen?</code>	Gibt die bisher generierte Karte auf dem Bildschirm aus
<code>Einzugsbereiche zeichnen?</code>	Zeichnet um jeden Kartenpunkt seinen Einzugsbereich (siehe Gleichung 5.20)

A.3 Das Kommando writepixel

Einstellbare Parameter

Objektkonturen in Pixelfile schreiben?	Schreibt die Konturpunkte der ausgewählten Objekte des aktuellen Radarbilds in ein File.
---	---

A.4 Das Kommando chartobj

Dieses Kommando erlaubt die Einstellung gewisser Schwellwerte für die Auswahl der Objekte des aktuellen Radarbilds. Nur Objekte, die alle drei Bedingungen erfüllen, gehen anschließend in die Karte ein.

Einstellbare Parameter

Minimaler sichtbarer Anteil	Minimaler Erstechoanteil eines Objektes
Minimale Objektflaeche	Schwellwert für die Objektfläche
Maximaler Abstand des Objektes	Objekte, die weiter entfernt sind, werden nicht berücksichtigt
Maximaler Abstand der Objektpunkte (*max_Distanz)	Maximale Entfernung von Konturpunkten, die in die Approximation eingehen sollen. Angege- ben wird ein Faktor für den maximalen Ab- stand des Objektes.

A.5 Das Kommando approx

Einstellbare Parameter

Minimale Abweichung	Schranke für die Abweichung zwischen Kontur und approximierender Linie, ab der die Linie nicht mehr weiter unterteilt wird
---------------------	--

A.6 Das Kommando chartkalman

Einstellbare Parameter

P-Matrixinit. SigmaXPos	Initialisierung der Diagonalelemente der Zustandskovarianzmatrix. Angegeben werden die Standardabweichungen: $\sigma_{x_{pos}}$
P-Matrixinit. SigmaYPos	$\sigma_{y_{pos}}$
P-Matrixinit. SigmaGeschwX	σ_{v_x}
P-Matrixinit. SigmaGeschwY	σ_{v_y}
Q-Matrixinit. SigmaXPos	Initialisierung der Systemkovarianzmatrix: $\sigma_{x_{pos}}$
Q-Matrixinit. SigmaYPos	$\sigma_{y_{pos}}$
Q-Matrixinit. SigmaGeschwX	σ_{v_x}
Q-Matrixinit. SigmaGeschwY	σ_{v_y}
R-Matrixinit. SigmaXPos	Initialisierung der Messungskovarianzmatrix: $\sigma_{x_{pos}}$
R-Matrixinit. SigmaYPos	$\sigma_{y_{pos}}$

A.7 Das Kommando chartmatch

Einstellbare Parameter

Einzugsbereichsgroesse einer Strecke (* min_dev)	Legt die Breite des Einzugsbereiches fest. Entspricht f aus Gleichung 5.22.
---	---

A.8 Das Kommando chartdel

Einstellbare Parameter

Bisher generierte Karte loeschen?	Löscht die Karte, so daß die Kartengenerierung beim aktuellen Radarbild von neuem beginnt.
--------------------------------------	--

Literaturverzeichnis

- [Borthwick et al. 94] BORTHWICK, S., H. DURRANT-WHYTE (1994): Simultaneous Localisation and Map Building for Autonomous Guided Vehicles. **Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems**, vol. 2, September 1994, 761-768.
- [Brammer et al. 85] BRAMMER, K., G. SIFFLING (1985): **Kalman-Bucy-Filter – Deterministische Beobachtung und stochastische Filterung**. Oldenburg-Verlag GmbH, München.
- [Brandes 95] BRANDES, M. (1995): **Entfernungsbestimmung markanter Punkte aus einem Fahrzeug mit Hilfe eines Stereokamerasystems**, Universität Stuttgart, Fachbereich Parallele und Verteilte Höchstleistungsrechner, Diplomarbeit.
- [Feiten et al. 92] FEITEN, W., G. LAWITZKY, I. LEUTHÄUSSER, M. MÖLLER, W. RENCKEN, R. BAUER (1992): Mobile low-cost Roboter in komplexen, quasistrukturierten Umgebungen. **Autonome Mobile Systeme**, November 1992, 234-243.
- [Gern 94] GERN, T. (1994): **Aufbau und Erprobung eines Transputersystems für die Fahrzeugführung in einer Kolonne**, Universität Stuttgart, Fachbereich Parallele und Verteilte Höchstleistungsrechner, Studienarbeit.
- [Gilles et al. 93] GILLES E.D., M. FAUL, U. KABATEK, M. SANDLER (1993): Automatisierung des Schiffsverkehrs auf Wasserstraßen. **Automatisierungstechnische Praxis 35**, R. Oldenburg Verlag, 543-552.
- [Gonzales et al. 92] GONZALES, R.C., R.E. WOODS (1992): **Digital Image Processing**. Addison-Wesley Publishing Company, 518-528.
- [Hermann 94] HERMANN, K. (1994): **Neuprogrammierung der radarbildverarbeitenden ProgrammROUTINEN des integrierten Navigationssystems für Binnenschiffe**, Universität Stuttgart, Fachbereich Systemdynamik und Regelungstechnik, Bericht.

- [Kanesalingam et al. 93] KANESALINGAM, C., S.J. DODDS, M.C.B. SMITH (1993): A Simple Mapping and Path Planning Algorithm for Mobile Robots. **IFAC Intelligent Autonomous Vehicles**, April 1993, 447-452.
- [Kistner 95] KISTNER, A. (1995): **Stochastische Systeme**. Institut A für Mechanik, Universität Stuttgart. Skript zur Vorlesung.
- [Knieriemen et al. 90] KNIERIEMEN, T., E. VON PUTTKAMER, J. ROTH (1990): Auswertung von Radarbildern mit Krümmungsalgorithmen im Vergleich mit Linienalgorithmen. **Autonome Mobile Systeme**, November 1990, 185-195.
- [Kurz 96] KURZ, A. (1996): Constructing Maps for Mobile Robot Navigation Based on Ultrasonic Range Data. **IEEE Transactions on Systems, Man, and Cybernetics**, vol. 26, no. 2, 233-242.
- [Kurz et al. 92] KURZ, A., L. CAMACHO (1992): Schritthaltende Erkennung von Umweltkonturen aus Infrarot-Radarbildern. **Autonome Mobile Systeme**, November 1992, 209-221.
- [Neul 92] NEUL, R. (1992): **Positionsbestimmung eines navigierenden Schiffes durch kartengestützte Radarbildverarbeitung**, Universität Stuttgart, Fachbereich Systemdynamik und Regelungstechnik, Dissertation.
- [Rosin et al. 95] ROSIN, PAUL L., GEOFF A.W. WEST (1995): Nonparametric Segmentation of Curves into Various Representations. **IEEE Transactions on Pattern Analyses and Machine Intelligence**, vol. 17, no. 12, 1140-1153.
- [Stuck et al. 94] STUCK, E.R., A. MANZ, D.A. GREEN, S. ELGAZZAR (1994): Map Updating and Path Planning for Real-time Mobile Robot Navigation. **Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems**, vol. 2, September 1994, 753-760.
- [Thrun et al. 96] THRUN, S., A. BÜCKEN (1996): **Learning Maps for Indoor Mobile Robot Navigation**, Carnegie Mellon University, Pittsburgh.