

Freie Universität Berlin
AG Künstliche Intelligenz, AG Neurobiologie

Studiengang Bioinformatik
Bachelorarbeit

Tracking von Radar-detektierten Flügen von Honigbienen

betreut durch
Prof. Dr. Raúl Rojas
Prof. Dr. Randolf Menzel
Tim Landgraf

von
Lutz Reiter

24. September 2009

Eigenständigkeitserklärung:

Hiermit erkläre ich, dass ich diese Bachelorarbeit selbständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen der Arbeit, die wörtlich oder sinngemäß aus Veröffentlichungen oder aus anderweitigen fremden Äußerungen entnommen wurden, sind als solche kenntlich gemacht.

Berlin, 24. September 2009

Lutz Reiter

Inhaltsverzeichnis

1	Einführung	3
1.1	Motivation	3
1.2	Navigation von Bienen	3
2	Das Radarsystem	4
2.1	Das Harmonische Radar	5
2.2	Präparation der Bienen	6
3	Implementierung	6
3.1	Problembeschreibung	6
3.2	Anforderungen	7
3.3	Lösungsplan	8
3.4	Der Tracking-Algorithmus	8
3.4.1	Erstellung der Maske	8
3.4.2	Binarisierung und Entfernung von Bildrauschen	9
3.4.3	Maximumsuche	9
3.4.4	Positionierung des Suchfensters	10
3.4.5	Herausfiltern doppelter Ortungen	10
3.5	Benutzeroberfläche	12
3.5.1	Use-Case Diagramm	12
4	Ergebnisse	13
4.1	Genauigkeit der Positionsbestimmung	13
4.1.1	Stabilität des Messpunkts	13
4.1.2	Messpunkt-Verschiebung	14
4.2	Test des Tracking Algorithmus gegen manuelles Tracking	15
4.3	Probleme	18
4.3.1	Zeitabstand zwischen den Radarframes	18
5	Diskussion	19
5.1	Future Work	20
A	Verwendete Programme und Bibliotheken	21
A.1	Konvertierung IplImage zu QImage	21
B	Bedienung des Programms	21
B.1	Hauptfenster	21
B.2	Tracking Parameter	23

Zusammenfassung

Ziel dieser Arbeit ist die Entwicklung einer Software, die es ermöglicht selbständig die Radarechos von Bienen auf Radarbildern zu verfolgen, um aus diesen Informationen eine Flug-Trajektorie zu erstellen. Dabei wird auf Schwierigkeiten der Radardetektion eingegangen und die Implementierung einer GUI gestützten Software erläutert.

1 Einführung

1.1 Motivation

Bienen sind in der Lage erstaunliche Navigationsleistungen zu vollbringen. Bisher war es nur möglich das Flugverhalten von Bienen über relativ kurze Distanzen zu beobachten. In sehr naher Entfernung lassen sich Bienen mit der Kamera oder dem bloßen Auge beobachten. Bei längeren Distanzen kann man zwar Ziel- und Abflugort der Bienen mittels mark-and-recapture-technique überwachen und somit Rückschlüsse auf das Flugverhalten ziehen. Die Bienen im direkten Flug zu beobachten war allerdings unmöglich. Mit Hilfe einer vom Team von Prof. Joe Riley entwickelten Methode zur Benutzung des harmonischen Radars [J. Riley, A. Smith 2002] ist es nun gelungen den Flug der Bienen kontinuierlich beobachten zu können. Zu diesem Zweck werden die Bienen mit einer kleinen Antenne (Transponder) versehen, welche das Radarsignal transformiert, zurückwirft und die Biene als Echo auf dem Radarschirm erscheinen lässt. Bei Feldexperimenten haben wir teilweise Flugzeiten von über 3 Stunden beobachtet. Aus diesem Grund ist es sinnvoll, die Radarbilder automatisch auszuwerten, da eine enorme Menge an Datenpunkten erzeugt wird. Ein weiteres Ziel dieser Arbeit ist es, die Präzision der Positionsbestimmung gegenüber dem manuellen Tracking zu verbessern.

1.2 Navigation von Bienen

„Wir stehen vor einem großen Bienenhaus. Zwanzig Völker sind nebeneinander untergebracht, ein Stock sieht aus wie der andere. Tausende von Arbeitsbienen fliegen ihre Tracht aus[...]. Wir fangen eine Biene ab, die eben nach Hause will, zeichnen sie durch einen Farbfleck, sperren sie in ein kleines Kästchen und lassen sie 2 km vom Bienenhaus entfernt fliegen, Ein Beobachter bleibt bei den Stöcken zurück - und berichtet uns, dass die gezeichnete Biene in ihren Stock geflogen ist, wenige Minuten, nachdem wir ihr die Freiheit wiedergegeben haben.“ [K. von Frisch, M. Landauer 1993, S. 142]

Für das Bienengedächtnis stellt dies eine erstaunliche Leistung dar. Denn setzt man etwa eine junge Biene den gleichen Bedingungen aus, findet diese nicht einmal auf 50 m Entfernung zu ihrem Stock zurück. Bienen müssen also erst ihre Umgebung kennenlernen. Sie tun dies ab ihrem 10. Lebenstag auf ihren *Orientierungsflügen* [K. von Frisch, M. Landauer 1993]. Wie orientiert sich die Biene nun? Sie besitzt dazu mehrere verschiedene Orientierungswerkzeuge:

- Orientierung am polarisierten Licht (Sonnenkompass)
- Erkennung von Landmarken (Waldränder, Straßen, Bäche,...)
- Entfernungsmessung über den *optischen Fluss* (Die Bewegung des Untergrunds bzw. der Umgebung)
- Wahrnehmung des Erdmagnetfelds (z.B. zum Wabenbau)

- Geruchssinn (z.B. Erkennung des Stock-Geruchs, Pheromone)

In welcher Form das Bienengedächtnis diese Informationen nun speichert, und zu einem späteren Zeitpunkt wieder abrufen kann, ist noch unklar. Dazu gibt es zwei Ansätze: Die *elementare* und *die kognitive Erklärung* [R. Menzel 2009].

Die elementare Erklärung betrachtet die einzelnen Navigationsleistungen getrennt und versucht diese auf einfache Routinen zurückzuführen. Nach dieser Theorie erstellt das Gedächtnis der Biene eine *egozentrische* Repräsentation der Umgebung. Dies bedeutet, dass die Biene nur Relationen zwischen ihrer Position und Objekten herstellt, nicht aber zwischen verschiedenen Objekten untereinander. Die Biene kann z.B. Landmarken mit bestimmten Verhaltensweisen verknüpfen. Trifft die Biene auf eine bekannte Landmarke, spielt sie ein gelerntes Verhaltensmuster ab und fliegt in eine bestimmte Richtung. Diese Art der Navigation nennt sich auch *Routen Navigation* [N. Fortin 2008]. Die Verhaltensmustern der Routen Navigation lassen sich relativ leicht erlernen, sie haben aber den Nachteil, dass die Informationen sehr unflexibel sind. Die Landmarken müssen immer in der gleichen Reihenfolge angefliegen werden und es müssen sowohl Routen für den Hin- als auch den Rückflug erlernt werden.

Ein anderes Hilfsmittel für die Orientierung ist die *Wegintegration* [L. F. Jacobs, F. Schenk 2003]. Sie erklärt, wie Bienen trotz eines kurvenreichen Flugs, auf dem direkten Weg zu ihrem Ausgangspunkt zurückfinden können. Dazu addiert sie die einzelnen Teilvektoren, deren Richtung sie anhand des polarisierten Lichts erkennt und deren Länge sie über den optical-flow messen kann. Auf dem Rückweg folgt sie dann einfach dem resultierenden Vektor in umgekehrter Richtung.

Die kognitive Erklärung geht (zumindest teilweise) von einer *allozentrischen* Repräsentation im Navigationsgedächtnis aus. Die Biene erstellt also von einigen Bereichen ihrer Umgebung eine *cognitive map*. Dazu nutzt sie die gleichen Informationen, welche ihr auch bei der egozentrischen Erfassung zur Verfügung stehen, nur setzt sie diese jetzt in Relation zu anderen Objekten ihrer Umgebung. Die einzelnen *Trajektorien* (Flugspuren) der Orientierungsflüge werden dabei zu einer Umgebungskarte integriert. Die Richtungsbestimmung über den Sonnenkompass und die Entfernungsmessung über den optical flow helfen ihr dabei sich in unbekanntem Terrain auch ohne Landmarken zu orientieren. Diese Theorie würde erklären, wieso Bienen, auch wenn man sie in weiter Entfernung zum Stock aussetzt, nach einer kurzen Orientierungsphase, in welcher sie ihre Position anhand von zuvor gespeicherten Landmarken und des Sonnenkompass bestimmen, auf direktem Weg zum Stock zurückkehren.

Die oben geschilderte von Karl von Frisch beobachtete Verhaltensweise könnte auch anhand eines umfangreichen Bildgedächtnisses erklärt werden [R. Menzel 2009], indem die Biene eine große Zahl von Routen zu verschiedenen Ausgangspositionen ihrer Umgebung speichert. Dies würde wiederum die elementare Erklärung unterstützen.

So gibt es bisher weder Belege für die eine, noch die andere Theorie. Die Beobachtung von Bienen mit dem harmonische Radar verspricht eine Klärung vieler offener Fragen in diesem Gebiet.

2 Das Radarsystem

Alle Radararten funktionieren im Prinzip gleich: Das Radargerät sendet gebündelte elektromagnetische Wellen aus - das *Primärsignal*. Dieses Signal wird von Oberflächen reflektiert, als *Sekundärsignal* vom Radargerät empfangen und bezüglich der Entfernung und des Winkels ausgewertet. Die Entfernung wird über die Messung der Laufzeit

des Radarsignals nach der Formel $t = \frac{2s}{c}$ (c = Lichtgeschwindigkeit) bestimmt. Das konventionelle Radar empfängt dabei die reflektierten Sekundärsignale aller Oberflächen in Strahlungsrichtung. Für die Ortung von Insekten, die sich in Bodennähe aufhalten ist dieses Verfahren daher ungeeignet, da es unmöglich ist, die Echos der Insekten aus Reflektionen von Vegetation und anderen Objekten herauszufiltern. Zur Ortung von Insekten wird daher das *harmonische Radar* eingesetzt.

2.1 Das Harmonische Radar

Um Objekte mit dem harmonischen Radar zu detektieren, müssen diese mit einem Transponder markiert werden. Dieser *passive* Transponder (siehe Abbildung 2) wirft das vom Radargerät ausgesandte Primärsignal als moduliertes Sekundärsignal einer anderen Frequenz zurück. Er benötigt dazu keine eigene Energiequelle, da er den Impuls des Primärsignals als Energiequelle nutzt um eine harmonische Frequenz (ein ganzzahliges Vielfaches der Ursprungs-Frequenz) des Primärsignals zu erzeugen.

Abbildung 1: Radar-Antenne



Die Empfangsantenne ist über der Sendeanterie angebracht. Der Empfängerkopf der Sekundäranterie (Empfang) hat einen halb so großen Durchmesser, wie der der Primäranterie (Senden) um die 2. harmonische Frequenz (doppelte Frequenz = halbe Wellenlänge) des Primärsignals detektieren zu können.

Da sich Primär- und Sekundärsignal in der Frequenz unterscheiden, benötigt das Radargerät neben der Sendeanterie eine gesonderte Empfangsantenne (siehe Abbildung 1). Diese Technik hat den entscheidenden Vorteil, dass Echos von normalen Oberflächen nicht empfangen und hauptsächlich die Echos des Transponders vom Empfänger registriert werden. Um eine möglichst hohe Sendeleistung des Transponders zu erreichen wird ein Impulsradar verwendet, dabei werden kurze hochenergetische Impulse ausgesandt. Diese Radarsysteme werden üblicherweise in der Seefahrt benutzt und sind deswegen sehr viel günstiger als z.B. ein FMCW-Radar¹.

¹Frequenz moduliertes Dauerstrich Radar

2.2 Präparation der Bienen

Der 10,5mg schwere Transponder besteht aus einer Diode, und einem gewundenen Stück Silberdraht (siehe Abbildung 2). Er wird über doppelseitiges Klebeband auf einem mit Schellack befestigten Nummernplättchen auf dem Thorax der Biene befestigt. Versuche haben gezeigt, dass die Bienen mit Transponder keine starken Beeinträchtigungen in ihrem normalen Verhalten zeigen. Setzt man sie in unbekannter Umgebung aus, konnten wir sie über mehrere Stunden in der Luft beobachten.

Abbildung 2: Biene mit Radar Transponder



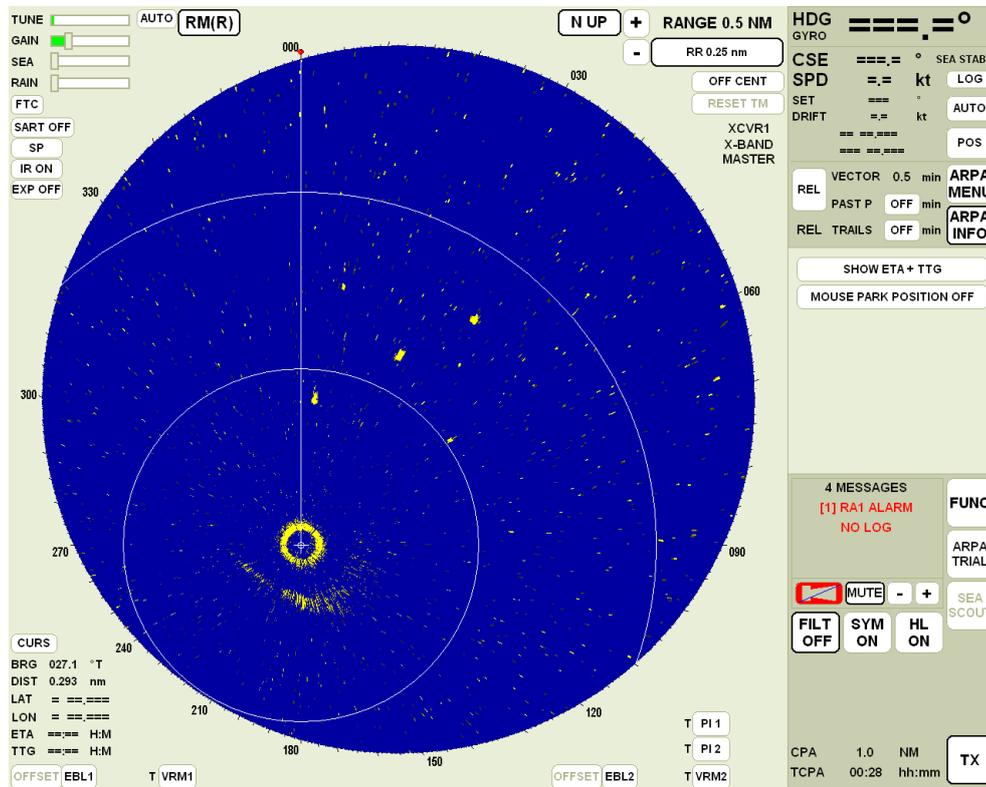
Der Transponder besteht aus einem Silberdraht, welcher in der Mitte zu einer Drosselspule gebogen ist, in welche eine *Schottky Low Barrier Diode* gelötet wird. Die Länge der beiden Silberstäbchen des Dipols beträgt jeweils 10mm. [B. Fischer, W. Haaß 2007] Das Primärsignal wird mit der Frequenz $f_1 = 9,4\text{GHz}$ ausgesandt, vom Transponder als 2. harmonischen Frequenz $f_2 = 18,8\text{GHz}$ zurückgeworfen und von der Sekundärantenne empfangen.

3 Implementierung

3.1 Problembeschreibung

Die Radardaten liegen als Einzelaufnahmen im BMP-Format vor. Dazu wird in regelmäßigen Abständen der Bildschirm des Radargeräts mittels eines Screenshot-Programms aufgenommen und über eine Netzwerkverbindung auf die Festplatte eines zweiten Rechners übertragen. Die Radarbilder sind teilweise stark verrauscht und beinhalten neben dem Radarschirm auch die GUI des Radarprogramms (siehe Abbildung 3). Durch Reflektionen im Nahfeld des Radars kommt es um den Radar Ursprung zu besonders starkem Rauschen. Die Radarechos verändern ihre Form in Abhängigkeit zur Signalstärke und zur Entfernung zum Radargerät. Falls die Biene zu tief fliegt oder im Gras sitzt, verschwindet das Echo komplett.

Abbildung 3: Radarbild



Um den blauen Radarschirm befindet sich die Bedienoberfläche des Radarsystems. Echos sind gelb markiert. Der Radar-Ursprung ist durch ein weißes Kreuz gekennzeichnet. Gut erkennbar ist hier das extreme Rauschen um den Radar Ursprung.

Die Umlaufzeit des Radars beträgt ca. 3 Sekunden, dabei wird der Radarschirm ständig aktualisiert. Da der zeitliche Abstand der Screenshots geringer ist, kommt es vor, dass ein identisches Echo auf zwei aufeinanderfolgenden Bildern auftaucht, obwohl sich die Biene schon weiterbewegt hat.

3.2 Anforderungen

Anforderungen an die Software:

- Einlesen der Bilddaten in chronologischer Reihenfolge
- Erkennung der Bienenechos
- Erstellung einer Trajektorie aus den Radardaten, welche auch die zeitlichen Abstände zwischen den Bildern enthält
- Graphische Darstellung der Trajektorie
- Ausgabe der Flugspur-Daten in Relation zum Radar Ursprung
- Anpassbarkeit des Algorithmus an unterschiedlich starkes Rauschen, unterschiedliche Echo- und Hintergrundfarben
- Graphische Benutzeroberfläche zur Bedienung des Programms

3.3 Lösungsplan

Die Radarechos werden schon vom Radargerät mittels eines Schwellwerts binarisiert, sie können leicht über die Farbinformationen extrahiert werden. Statische Echos, welche durch Reflektionen von Zäunen oder anderen metallischen Objekten entstehen, sowie das starke Rauschen im Nahfeld müssen maskiert werden. Da sich Form und Größe der Echos mit jedem Bild ändert, lassen sich die Echos nur über ihre Größe und relative Position zum vorherigen Auftauchen bewerten. Aus diesem Grund wird ein dynamisches quadratisches *Suchfenster* benutzt, welches zwischen den einzelnen Frames vergrößert wird. Im letzten Schritt müssen aufgrund der Asynchronität von Radargerät und Screenshot-Programm doppelte Ortungen entfernt werden.

3.4 Der Tracking-Algorithmus

Der Algorithmus führt folgende Schritte zur Erkennung der Bienenposition durch:

1. Erstellung einer Maske
2. für jedes Bild:
 - (a) Radarbild binarisieren
 - (b) Maske auf Binärbild anwenden
 - (c) Entfernen von Rauschen
 - (d) Maximumssuche mit Schwellwert im Suchfenster
 - (e) Suchfenster neu positionieren
3. Herausfiltern doppelter Ortungen

3.4.1 Erstellung der Maske

Nach dem Einlesen der Bilder I_i werden 10 zufällige Bilder zum Erstellen der Maske verwendet. Es wird eine Nullmatrix M in Bildgröße erstellt. Jedem Element der Matrix M wird für $i=1,\dots,10$ nach Formel (1) ein Wert zugeordnet

$$M(x, y) = \begin{cases} M(x, y) + 1, & \text{falls } I_i(x, y) \neq \text{Hintergrundfarbe} \\ M(x, y), & \text{sonst} \end{cases} \quad (1)$$

Anschließend wird aus M ein Binärbild als Maske erzeugt, dessen Pixel den Wert 1 erhalten, falls $M(x, y) < t$ ist. t ist dabei ein Schwellenwert, welcher die Anzahl der Bilder festlegt, in denen ein Pixel (\neq Hintergrundfarbe) vorkommen muss um maskiert zu werden. Im anderen Fall, wenn $M(x, y) \geq t$, erhält der Pixel den Wert 0. Wird diese Matrix mit einem Graubild elementweise multipliziert, bleiben nur die Pixel, welche den Masken-Wert 1 besitzen sichtbar.

Die Maske entfernt nicht nur statische Echos und Störstrahlung vom Radarschirm, sondern auch andere überflüssige Bildinformationen (Koordinatensystem, Rahmen, zusätzliche Informationen), welche auf den Radar-Screenshots vorhanden sind (siehe Abbildung 5).

3.4.2 Binarisierung und Entfernung von Bildrauschen

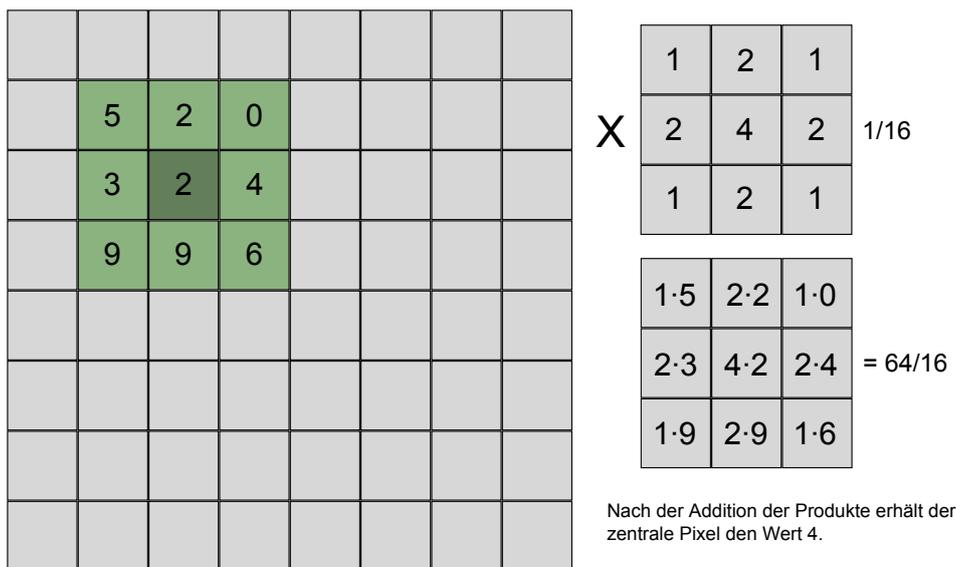
Im ersten Schritt der Tracking-Schleife wird nun aus dem Radarbild ein Binärbild erzeugt, indem alle Pixel, welche der Echo-Farbe entsprechen den Wert 1 erhalten, alle anderen Pixel werden auf 0 gesetzt. Dieses Bild wird nun mit der Maske verundet, ein Grauwert-Bild erzeugt und mittels eines Gaußfilters weichgezeichnet.

$$\frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

3x3 Gauß Kernel

Dazu wird der Gauß-Kernel über jedem Pixel des maskierten Ausgangsbildes zentriert. Der Wert des Ausgabebildes errechnet sich dann aus der elementweisen Multiplikation des Gauß-Kernels mit den darunterliegenden Pixelwerten und der anschließenden Addition dieser Produkte (siehe Abbildung 4).

Abbildung 4: Korrelation mit 3x3 Gauss Kernel



Durch die Symmetrie des Gauß-Kernels wird die Faltung auch eine Korrelation, aus diesem Grund sollte die Größe des Kerns der Größe des Echos entsprechen. Denn der Gaußfilter dient neben der Entfernung des Bildrauschens auch als Bewertungsfunktion. Befindet sich ein großes Echo innerhalb des Filterbereichs, erhält der zentrale Pixel einen höheren Wert.

3.4.3 Maximumsuche

Anschließend wird eine Maximumsuche im Suchfenster durchgeführt. Falls das Maximum über einem Schwellenwert t liegt, wird der Mittelpunkt des Echos mittels *running average* bestimmt. So findet der Algorithmus auch bei Echos, welche größer als der

Filter-Kernel sind den Mittelpunkt. Das *running average* ist der ungewichtete Mittelpunkt \vec{m} aller Pixel mit dem Maximalwert. Er lässt sich kumulativ nach folgender Formel errechnen:

$$\vec{m}_{i+1} = \frac{\vec{p}_{i+1} + i\vec{m}_i}{i+1}$$

wobei \vec{p}_{i+1} jeweils ein neuer Punkt mit einem Pixelwert gleich dem Maximum ist.

Falls der Pixelwert des errechneten Punktes größer als t ist, wird der Punkt zusammen mit seinem Pixelwert gespeichert.

3.4.4 Positionierung des Suchfensters

Die Positionierung des Suchfensters hängt von 3 Parametern ab: die maximale Flugeschwindigkeit der Biene v_{max} , der Zeitpunkt der letzten Detektion der Biene t_{i-1} und einem Maximalwert s_{max} für die Größe des Fensters. Das Suchfenster wird nach dieser Regel skaliert:

$$s_i = \min((t_i - t_{i-1})v_{max}, s_{max})$$

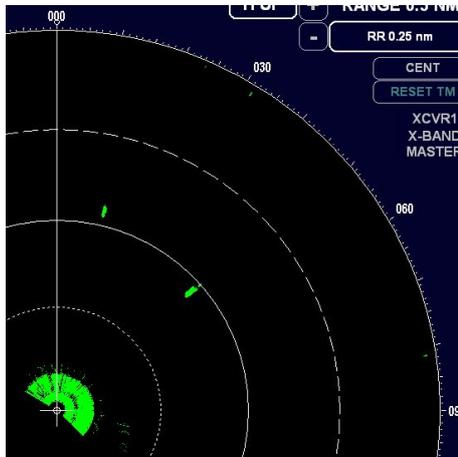
Wobei t_i den momentane Zeitpunkt und s die halbe Kantenlänge des quadratischen Suchfensters darstellt.

Die Beschränkung auf eine maximale Suchfenstergröße ist sinnvoll, da davon auszugehen ist, dass eine Biene, welche über einen längeren Zeitraum nicht detektiert wurde, sich z.b. auf eine Pflanze gesetzt oder einem anderen Objekt niedergelassen hat, aus diesem Grund nicht mehr vom Radar detektiert wird und an der gleichen Position bleibt.

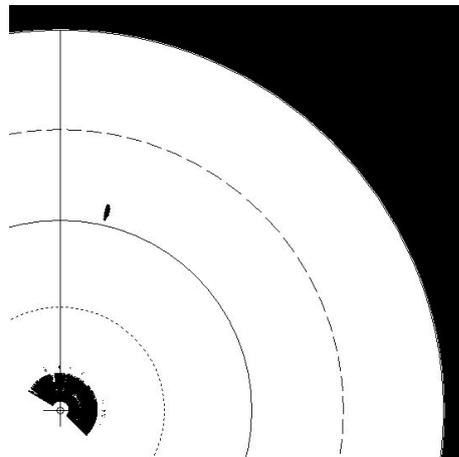
3.4.5 Herausfiltern doppelter Ortungen

Das Problem doppelter Ortungen tritt auf, da das Radargerät nicht mit dem Screenshot Programm synchronisiert wird. Das Radargerät hat eine Umlaufzeit von 3 Sekunden, während das Screenshot Programm Bilder im 1-3 Sekunden Takt erzeugt. Aus diesem Grund tauchen einzelne Echos in mehrfacher Ausführung in den Positionsdaten auf. Da die Positionsdaten allerdings immer mit ihrer Echobewertung (errechnet durch den Gaußfilter aus Form und Größe des Echos) gespeichert werden und diese sich mit jeder Radarumdrehung ändert, lassen sich diese durch das Löschen sich mehrfach wiederholender Routenpunkte bereinigen. Durch den Abgleich der Bewertung ist sehr unwahrscheinlich, dass ein falscher Datenpunkt gelöscht wird, auch wenn sich die Biene zwischen den Frames nicht bewegt hat (siehe Abbildung 6).

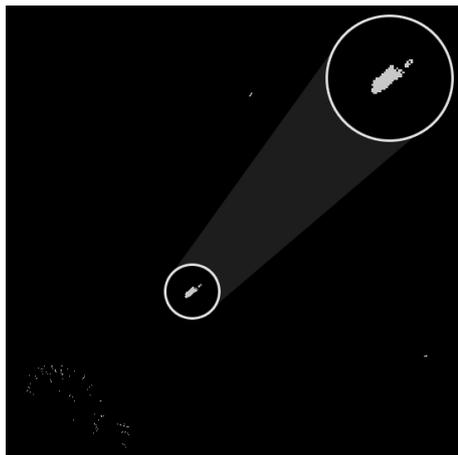
Abbildung 5: Die Schritte des Tracking Algorithmus



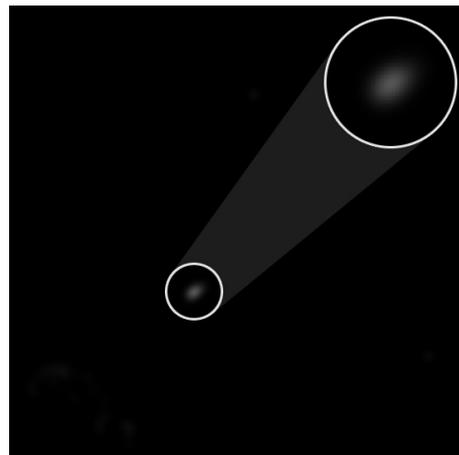
Ausgangsbild: Das linke Echo ist das statische Echo eines Zaunes. Das rechte Echo ist eine fliegende Biene.



Maske: Das statische Echo wird maskiert (schwarz), die Bienen-Position bleibt unmaskiert (weiß).



maskiertes Ausgangsbild: das statische Echo und die Reflektionen im Nahfeld um den Radar-Ursprung wurden entfernt. Das Bienenecho bleibt sichtbar.



weichgezeichnetes Bild: Durch Anwendung des Gauß-Filters wird Hintergrundrauschen entfernt. Der Mittelpunkt des Bienenechos erhält den Maximalwert.



markiertes Echo: Die Maximumsuche findet den Mittelpunkt des Bienenechos (Kreis-Markierung). Das Suchfenster ist hier als weißes Rechteck eingezeichnet.

Abbildung 6: Filterung doppelter Ortungen

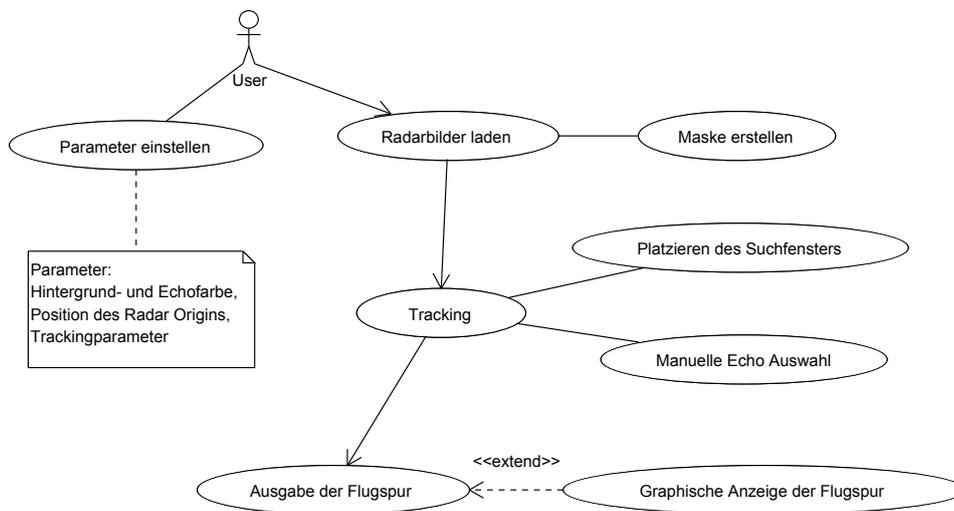
// ungefilterte Daten				// gefilterte Daten			
time	pos_x	pos_y	value	time	pos_x	pos_y	value
0	-28	121	79	0	-28	121	79
1	-28	121	79	-			
2	-38	104	46	2	-38	104	46
3	-47	100	91	3	-47	100	91
4	-47	100	91	-			
5	-47	100	91	-			
6	-36	120	78	6	-36	120	78
7	-36	120	78	-			

3.5 Benutzeroberfläche

Die GUI des Programms wurde mit der Klassenbibliothek QT 4.3 erstellt. QT bietet eine Multi-Platform Bibliothek und unterstützt verschiedene Programmiersprachen. Auf der Trolltech Website kann man eine sehr umfangreiche Dokumentation² zu den verschiedenen Modulen QTs finden.

3.5.1 Use-Case Diagramm

Abbildung 7: Use-Case Diagramm



Der Programmablauf vom Einlesen der Radarbilder bis zur Ausgabe der Flugspur. Die manuelle Auswahl der Echos und des Suchfensters ist nur nötig, wenn das Programm das Bienecho verliert oder es zu klein ist.

Um einen Tracking Vorgang zu starten, müssen zunächst die Parameter konfiguriert werden. Nun können die Radarbilder geladen werden. Danach wird die Maske erstellt. Anschließend kann der Tracking-Algorithmus gestartet werden. Sollte das Bienecho zu klein werden, kann es per Hand detektiert werden. Falls das Echo aus dem

²QT Dokumentation: <http://doc.trolltech.com/4.3/index.html>

Suchfenster herauspringt, muss das Suchfenster manuell justiert werden. Nach abgeschlossenem Tracking kann die Trajektorie graphisch angezeigt und die Route in einer Datei gespeichert werden.

4 Ergebnisse

4.1 Genauigkeit der Positionsbestimmung

Die Genauigkeit der Radar-Ortung hängt von mehreren Faktoren ab: die Auflösung des Radarsystems, die Auflösung der Radarbilder, die Größe des Bienenechos und die Wahl des Messpunktes.

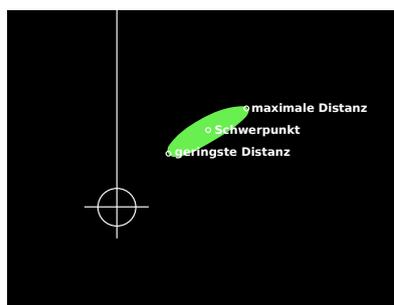
Während die Genauigkeit des Radarsystems sich nur durch Messung auf dem Feld bestimmen lässt, kann man die Genauigkeit des Algorithmus durch Betrachtung statischer Echos auf vorhandenen Radarbildern feststellen. Dabei wird die maximale Genauigkeit durch die Auflösung der Radarbilder begrenzt. Der Radius des Radarschirms beträgt 460 Pixel und umfasst eine halbe nautische Meile ($1 \text{ nm} = 1849,126 \text{ m}$). Dies entspricht einer Auflösung von 2,00 Metern pro Pixel.

4.1.1 Stabilität des Messpunkts

Um den stabilsten Messpunkt (Punkt des Blobs, der die Position des Echos definiert) zu bestimmen wurden Bilderserien mit statischen Echos in verschiedenen Entfernungen ausgewertet (5 unterschiedliche Echos in 3 Bilderserien mit jeweils mindestens 150 Einzelbildern) und deren Bewegung getrackt. Es kommen 3 definierbare Messpunkte in Frage:

1. Der mit dem Gaußfilter und running average bestimmte Schwerpunkt des Echos,
2. Punkt des Blobs mit minimalem Abstand zum Radar
3. Punkt des Blobs mit maximalem Abstand zum Radar

Abbildung 8: Messpunkte



Die Messung ergab, dass der Schwerpunkt der stabilste Punkt zur Bestimmung der Position ist (siehe Tabelle 1 und 2). Allerdings repräsentiert dieser nicht unbedingt die absolute Position der Biene, dazu wäre der Punkt mit minimalem Abstand zum Ursprung besser geeignet (siehe Abschnitt 4.1.2). Da die relative Position der Echos für die Auswertung eine bedeutendere Rolle spielt, wird im folgenden der stabilere Schwerpunkt verwendet.

Tabelle 1: Stabilität der Messpunkte in radialer³ Richtung

Messpunkt	Standardabweichung	Min	Max
1. Schwerpunkt	1,08m	-5,70m	+3,80m
2. Punkt mit min. Entfernung	1,78m	-8,49m	+7,30m
3. Punkt mit max. Entfernung	1,43m	-8,40m	+10,04m

Min/Max = Minimale und Maximale Abweichung vom Mean

Tabelle 2: Stabilität der Messpunkte in angularer⁴ Richtung

Messpunkt	Standardabweichung	Min	Max
1. Schwerpunkt	0,130°	-0,306°	+0,336°
2. Punkt mit min. Entfernung	0,393°	-0,460°	+0,475°
3. Punkt mit max. Entfernung	0,219°	-0,378°	+0,524°

Min/Max = Minimale und Maximale Abweichung vom Mean

4.1.2 Messpunkt-Verschiebung

Radarechos des harmonischen Radars haben die Eigenschaft, dass Echos, die näher am Radar liegen, in radialer Richtung stärker auseinandergezogen werden. Diese Eigenschaft wird durch die keulenförmige Ausbreitung und des dem damit einhergehenden Energieverlust des Primär- und Sekundärsignals verursacht. Der Transponder transformiert das Primärsignal und sendet das Sekundärsignal horizontal in alle Richtungen ab. Befindet sich der Transponder in naher Entfernung zum Radargerät, ist das Signal wesentlich stärker und wird teilweise auch über Umwege (Boden, Vegetation, andere Objekte) zurückreflektiert bzw gebeugt. Da diese Strahlen eine vergleichsweise zum direkten Weg längere Strecke zurücklegen und später im Radar eintreffen, wird das Echo auseinandergezogen. Bei größerer Entfernung sind diese Signale zu schwach und werden bei der Binarisierung des Radarbildes herausgefiltert (siehe Abbildung 9).

Aus dieser Eigenschaft lässt sich die *reale Distanz*⁴ eines Echos zum Radar-Ursprung als der minimale Abstand des Echo-Blobs zum Ursprung bestimmen, da dieser dem direkten Weg (Radar→Objekt→Radar) ohne zusätzlicher Umwege entspricht.

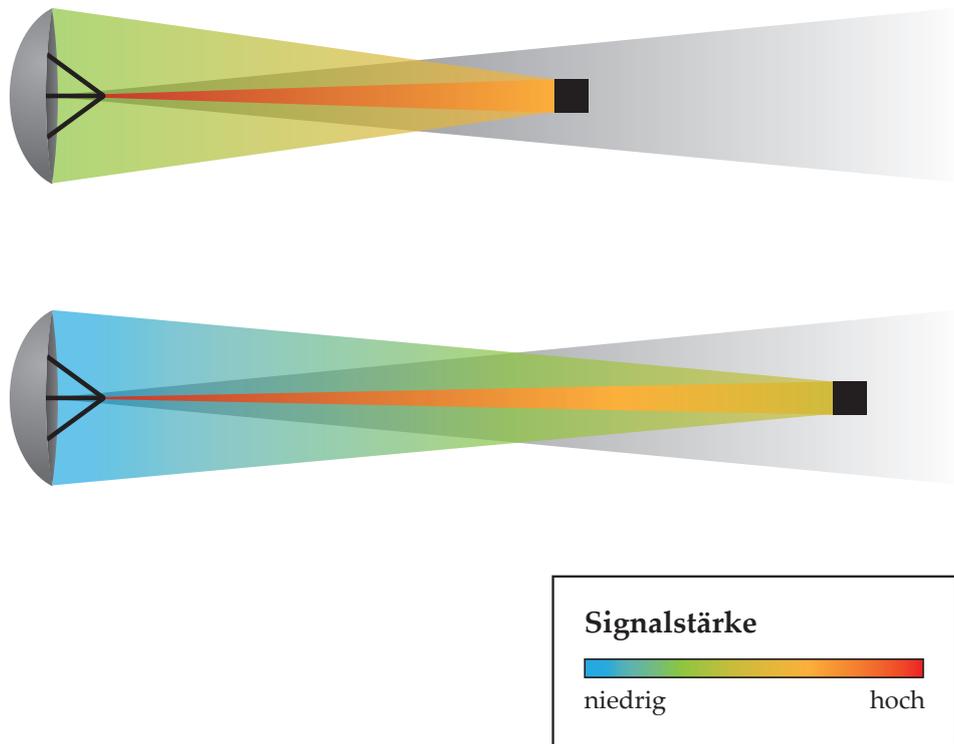
Da der Echo-Blob also bei geringerer Distanz zum Radar-Ursprung gestreckt wird, und sich damit die Distanz vom Messpunkt (der Schwerpunkt des Echos) zur realen Position (der Pixel des Blobs mit dem geringsten Abstand zum Radar Origin) der Biene vergrößert, ließe sich also eine Funktion finden, welche die Echo-Position in Abhängigkeit von der Entfernung zum Radar anpasst. Dazu wurde die Differenz zwischen dem Schwerpunkt und des Punktes mit minimaler Entfernung zum Radar-Ursprung, in Abhängigkeit zur Entfernung zum Radar gemessen. Leider konnten unter den bisher aufgenommenen Radarbildern nicht genügend unterschiedliche statische Echos gefunden werden, so dass sich eine solche Funktion nicht eindeutig bestimmen ließ (siehe Abbildung 10).

³Abstand zum Ursprung im Polarkoordinatensystem

³Winkel im Polarkoordinatensystem

⁴Die durch GPS-Messungen auf dem Feld bestimmbare Distanz

Abbildung 9: Reflektion der Radarstrahlen aus verschiedenen Distanzen



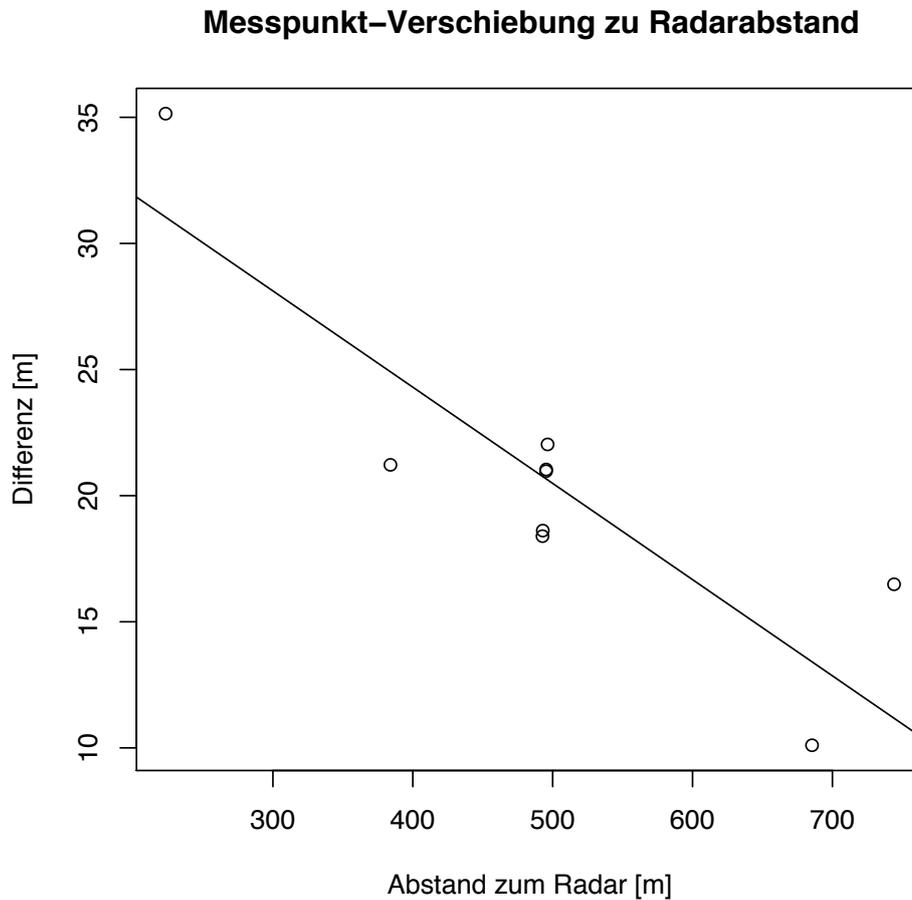
Abnahme der Signalstärke wegen des kegelförmigen Charakters und Streuung des Radarimpuls: Das Primärsignal wird von der Antenne abgestrahlt und breitet sich kegelförmig aus (grauer Strahl). Da sich das Primärsignal kegelförmig ausbreitet, nimmt die Signalstärke bei steigender Distanz zum Radargerät ab und es kommt nur ein Bruchteil des Primärsignals am Transponder an (roter Strahl). Der Transponder transformiert das Signal und strahlt es Horizontal in alle Richtungen ab, wobei nur ein kleiner Teil des abgestrahlten Sekundärsignals im Radarempfänger ankommt (grün bzw. blauer Strahl).

4.2 Test des Tracking Algorithmus gegen manuelles Tracking

Die manuell verfolgten Flugspuren wurde während des Experiments mittels eines in die Radarsoftware integrierten Maus-Positions Loggers erstellt. Dabei musste während der gesamten Dauer des Experiments eine Person anwesend sein, um die Maus jeweils auf die neue Echo Position zu bewegen. Die Maus-Position wurde über die serielle Schnittstelle des Radargeräts in Form von Polarkoordinaten ausgelesen, wobei die Präzision sich auf $1/100$ Nautische Meile (entspricht 18,49 m) und $1/10$ Grad beschränkte. Aus diesem Grund ist ein Rasterungseffekt auf den manuellen Flugspuren zu erkennen (siehe Abbildung 11).

Das automatische Tracking muss je nach gewünschter Präzision überwacht werden. Hohe Schwellwerte für die Maximumsuche erkennen zwar die Flugspuren zuverlässig, allerdings kommt es häufig vor, dass das Echo zu klein wird um automatisch erkannt zu werden, ohne dass es zuverlässig vom Rauschen unterschieden werden könnte. In diesem Fall muss entweder der Schwellwert gesenkt werden oder das Echo manuell ausgewählt werden.

Abbildung 10: Messpunkt-Verschiebung zu Radar-Abstand

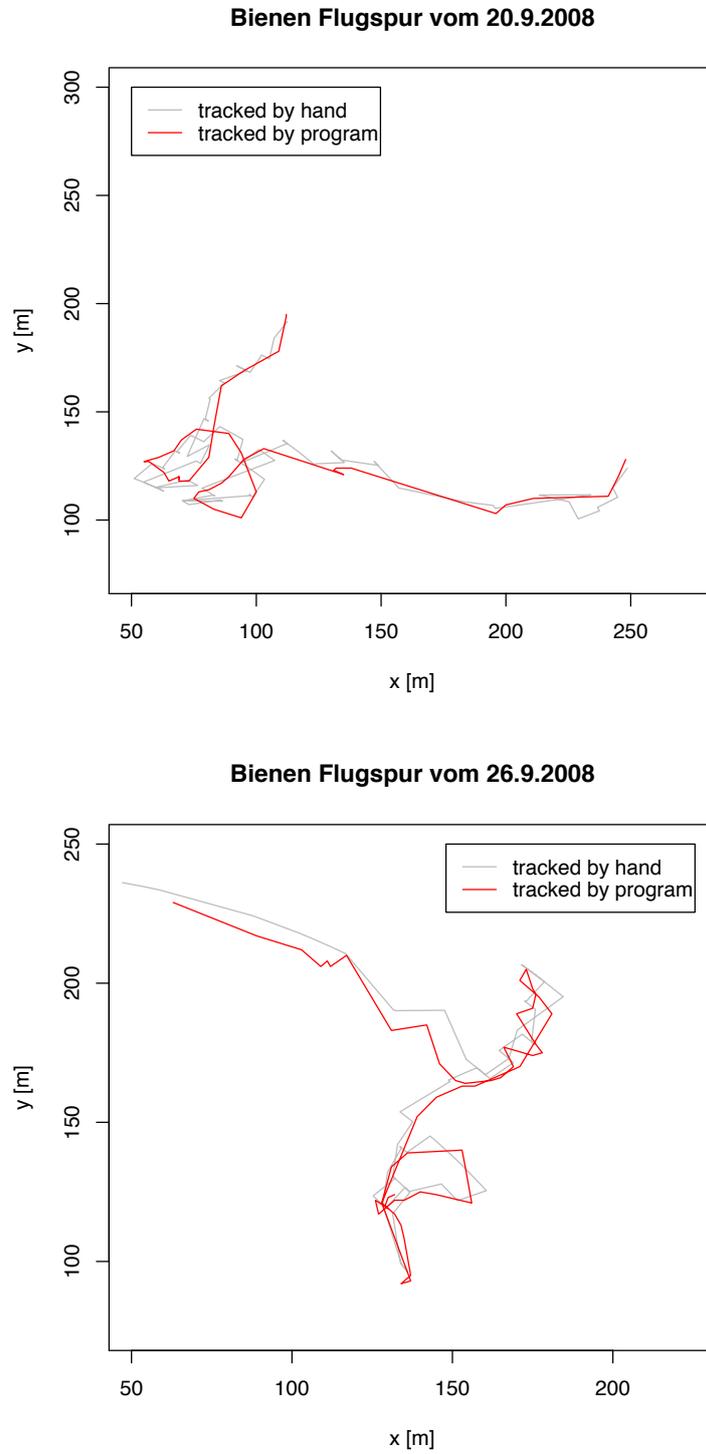


Die Grafik zeigt die Differenz zwischen dem Abstand des Schwerpunkts und des minimalen Abstands eines Blobs zum Radar Origin (y -Achse) zur Entfernung des Schwerpunkts (x -Achse) vom Radar Origin. Zusätzlich wurde die Regressionsgerade eingezeichnet.

Beide Flugspuren sind relativ deckungsreich (siehe Abbildung 11), wobei man bei den manuellen Flugspuren einen Rasterungseffekt erkennt. Desweiteren können die Bienechos je nach Entfernung zum Radar Ursprung eine Größe von 30×30 Pixeln haben. Das entspricht einer realen Ausdehnung von 120×120 m. Den Schwerpunkt zuverlässig und präzise per Hand zu bestimmen ist hier unmöglich.

Die manuell erstellten Flugspuren eignen sich wegen der verwendeten Hardware (im Radargerät integrierter Track-Ball) und der unpräzisen Übertragung der Positionen nur eingeschränkt zur Validierung.

Abbildung 11: Flugspuren zweier Experimente



Zwei Bienen-Flugspuren aufgenommen im September 2008.

4.3 Probleme

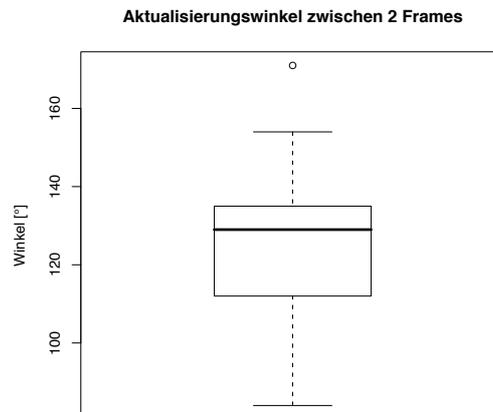
4.3.1 Zeitabstand zwischen den Radarframes

Die Radarbilder liegen im BMP-Format vor und wurden in regelmäßigen Abständen von einem Screenshot-Programm auf dem Radar-Rechner aufgenommen. Die Screenshots wurden direkt über eine Netzwerkverbindung auf die Festplatte eines zweiten Rechners geschrieben. Es wurden keine Log-Dateien mit den Aufnahmezeitpunkten der einzelnen Bilder gespeichert. Der Timestamp der einzelnen Bilddateien hat nur eine Genauigkeit von einer Sekunde und ist wegen der Verzögerung beim Schreiben über das Netzwerk sehr ungenau. Da es einige Unstimmigkeiten (sehr große Sprünge eines Bienen-Echos zwischen 2 Frames) in den Trajektorien gab, musste überprüft werden, ob das Screenshot Programm zuverlässig im vorgegeben Abstand von einer Sekunde Bilder aufgenommen hat.

Dazu wurden Bildserien mit besonders großem Radar-Rauschen ausgewertet. Das Radarbild wird auf dem Radarrechner kontinuierlich aktualisiert und die Umlaufzeit beträgt 3 Sekunden. Diese Eigenschaft lässt sich nutzen um den Winkel, welcher zwischen 2 Frames auf dem Radarschirm neu gezeichnet wurde, zu bestimmen. Dazu muss das Differenzbild zweier aufeinanderfolgender Frames gebildet werden. Auf dem Differenzbild erscheinen nur die aktualisierten Pixel, während alle statischen Pixel gelöscht werden. Aufgrund des Rauschens erhält man eine große Anzahl von aktualisierten Pixeln. Diese Pixel werden nun jeweils durch eine Gerade mit dem Ursprung verbunden, deren Winkel in Relation zur Polarachse gemessen wird. Anschliessend findet man die beiden Geraden mit der größten Winkeldifferenz und erhält den Aktualisierungswinkel (siehe Abbildung 13).

Die Messung des Aktualisierungswinkel (2 verschiedene Experimente mit jeweils 200 Bildern) ergab einen Mean von 125° und eine Standardabweichung von 16° , ein Maximum von 154° und ein Minimum von 98° . Ein Winkel von 125° entspricht bei einer Umlaufzeit von 3 Sekunden einem durchschnittlichem Abstand von 1041 ms zwischen den einzelnen Frames, mit einer Standardabweichung von 133 ms (siehe Abbildung 12).

Abbildung 12: Boxplot der Aktualisierungswinkel

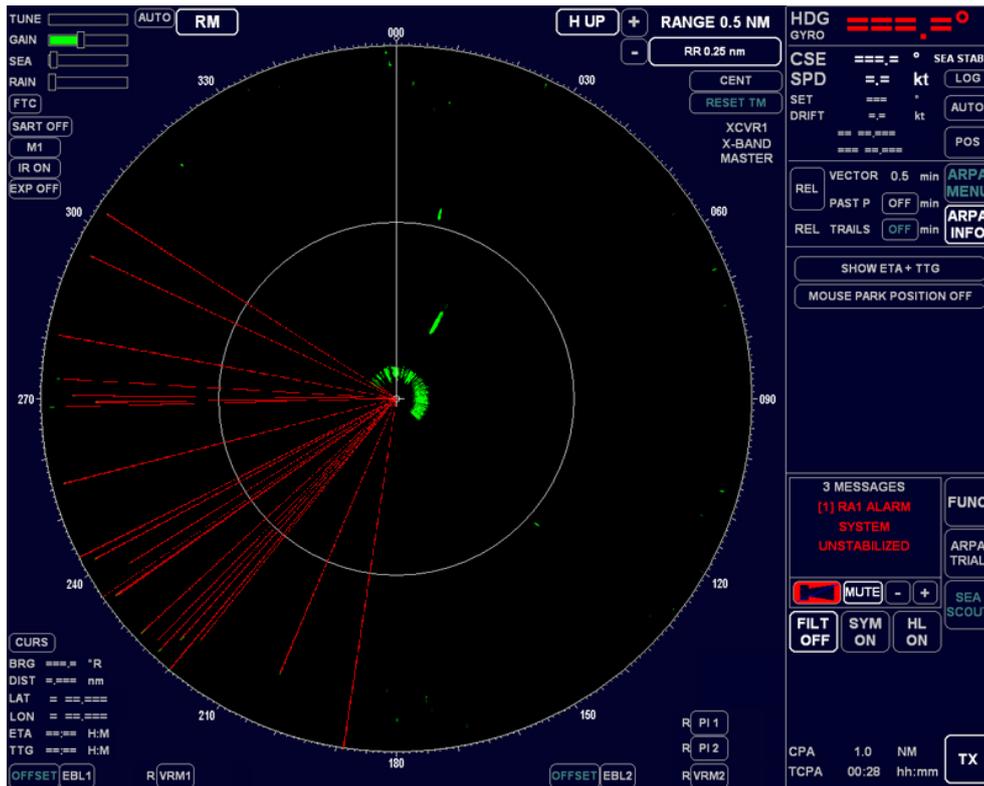


Dabei muss berücksichtigt werden, dass diese Messungen auf dem Hintergrundrau-

schen der Bilder basieren und der wirkliche Aktualisierungswinkel immer größer bzw. gleich dem gemessenen Winkel ist. Ein weiterer Faktor, der bei dieser Überprüfung eine Rolle gespielt hat, ist die Variabilität der Umlaufzeit des Radars. Bei starkem Wind weicht diese vom Normalwert ab.

Abschließend lässt sich feststellen, dass unter den gemessenen 400 Bildern kein Bild ausgelassen wurde. Das Screenshot Programm arbeitet also relativ zuverlässig, wenn auch nicht sehr exakt.

Abbildung 13: Aktualisierungswinkel zwischen zwei Frames



Die aktualisierten Pixel eines Frames sind hier durch rote Linien mit dem Radar-
Origin verbunden.

5 Diskussion

Die Software konnte die Erstellung der Trajektorien aus den Radardaten deutlich vereinfachen. Es ist nun nicht mehr nötig, während der kompletten Laufzeit des Experiments das Radarecho der Biene per Maus-Cursor verfolgen zu müssen. Die Person am Radar muss während des Experiments neben dem Verfolgen der Bienenposition zusätzlich Funkkontakt mit den Mitarbeitern im Feld halten und Protokoll führen.

Mit der Software können eine halbe Stunde Radardaten in ca. 10 Minuten ausgewertet werden. Ein weiterer Vorteil ist die Möglichkeit mehrere Bienen mit Transponder gleichzeitig fliegen zu lassen. Die Bienenspuren können später durch den Suchfenster-Mechanismus getrennt voneinander getrackt werden. Für den Fall, dass sich die beiden Bienenrouten kreuzen, muss das Echo manuell ausgewählt werden.

Desweiteren konnte die Präzision des Trackings erhöht werden. Durch die Bestimmung des Schwerpunkts der Echo-Blobs mittels des Gauß-Filters, sowie des running-average-Verfahrens ist sichergestellt, dass immer der Mittelpunkt des Blobs gefunden wird. Bei einer Ausdehnung der Echos von bis zu $30 * 30$ Pixeln ist es kaum möglich den Schwerpunkt per Hand zu bestimmen.

Die Visualisierung der getrackten Route erlaubt eine schnelle Kontrolle der Flugspuren und eine Übersicht über erste Ergebnisse des Experiments. So können die Flugrouten schon wenige Minuten nach dem Experiment präsentiert werden.

Allerdings ist es nicht gelungen die Bienen unüberwacht Tracken zu können. Auf einigen Bildern ist das Rauschen zu stark und die Identifizierbarkeit der Echos zu gering. Eine Möglichkeit die Sensitivität zu erhöhen stelle ich im folgenden Abschnitt vor.

5.1 Future Work

Wie schon in Abschnitt 4.1.2 beschrieben ließe sich die Genauigkeit der Positionsbestimmung erhöhen, wenn man eine Funktion finden würde, welche die Größe des Echos bzw. die Verschiebung der realen, mit GPS ermittelbaren Position mit der Entfernung zum Radar Ursprung in Relation setzt. Diese Funktion könnte man ermitteln, indem man mit einem Transponder in der Hand vom Radar aus startet und sich von ihm entfernt. Dabei muss die zurückgelegte Entfernung gemessen werden und sein Radarecho an jedem Punkt aufgezeichnet werden. Anschließend muss die Person auf den Radarbildern getrackt werden und ihre Position auf den Radarbildern mit ihrer aufgezeichneten realen Position verglichen werden. Die Entfernungsmessung könnte am einfachsten per GPS erfolgen.

Um die Sensitivität des Algorithmus zu verbessern, könnte man an der Maximumsuche Funktion ansetzen. Diese Funktion arbeitet mit einem festen Schwellenwert (siehe Abschnitt 3.4.3). Diesen kann man in Abhängigkeit zum Bildrauschen anpassen. Werden für einige aufeinanderfolgende Frames nur eindeutige Kandidaten für die Position der Biene und nur wenige, kleine falsche Echos im Suchfenster gefunden, kann der Schwellenwert gesenkt werden. Sobald wieder größere Falsch-Echos auftauchen, muss er erhöht werden.

Um die Arbeit des überwachten Trackings zu Erleichtern, wäre eine visuelle Kennzeichnung aller möglichen (Positions-)Kandidaten im Suchfenster sinnvoll. Falls mehrere mögliche Kandidaten gefunden werden, wird das Tracking gestoppt und der richtige Kandidat per Hand ausgewählt.

Die entscheidende Verbesserung um die Auswertung der Flugspuren zu vereinfachen wäre jedoch eine Aufzeichnung der genauen Bildaufnahme-Zeitpunkte in Millisekunden. Diese ließen sich beispielsweise als Metadaten im EXIF-Standard vom Screenshot Programm speichern und vom Tracking Programm auslesen. Momentan wird angenommen, dass zwischen allen Frames einer Bildserie die gleichen Zeitabstände liegen. Zwar ist der Fehler der Zeitabstände über wenige Frames betrachtet relativ gering (siehe 4.3.1), allerdings dürfte er sich über einige hundert Frames akkumulieren und somit keine zuverlässigen Angaben über die genaue Flugzeit aus den Tracking-Daten zulassen.

A Verwendete Programme und Bibliotheken

Das Programm wurde in C++ implementiert. Im ersten Schritt wurden unter Benutzung von Intels OpenCV Bibliothek der Kern der Software, der Tracking Algorithmus, geschrieben. Im zweiten Schritt wurde die GUI unter Verwendung der QT-Bibliothek konzipiert. Als Entwicklungsumgebung wurde Eclipse CDT⁵ mit QT-Plugin verwendet.

Intels **OpenCV**⁶ ist eine weitverbreitete open-source Bibliothek zur Bildverarbeitung und zum maschinellen Sehen. Sie ermöglicht unter anderem das Einlesen von Bilddaten und stellt einige sehr schnelle Filteralgorithmen zur Verfügung.

QT⁷ ist eine populäre Klassenbibliothek zur GUI-Programmierung, die unter der GPL veröffentlicht wird. Es existieren Versionen für C++, Python, Java, Ruby, PHP und Perl, außerdem unterstützt sie die Grafikplattformen X11, Windows und Mac OS. QT verwendet zur Erzeugung des C++ Codes zusätzlich den Präprozessor *qmake* um einige nicht standardkonforme Erweiterungen der C++ Sprache zu ermöglichen. So verwendet es zur Kommunikation der einzelnen GUI Elemente das Signal- und Slot-Prinzip.

A.1 Konvertierung IplImage zu QImage

Die Bilddaten werden über die Funktion `cvLoadImage()` der OpenCV Bibliothek in die Datenstruktur `IplImage` eingelesen. Um diese Daten in der GUI anzeigen zu können, werden sie mit der Funktion `convertToQImage()` in ein Objekt vom Typ `QImage` konvertiert.

```
1 //works only for 24 bit rgb 3 channel pictures
2 QImage* convertToQImage(IplImage *image) {
3     int height = image->height;
4     int width = image->width;
5     uchar* data = (uchar*)image->imageData;
6     QImage* qimg = new QImage( data,width,height,
7                               QImage::Format_RGB888);
8     *qimg = qimg->rgbSwapped();
9     return qimg;
10 }
```

B Bedienung des Programms

Um einen Tracking Vorgang zu starten, müssen zunächst die Bilder geladen werden (*Load Images*). Danach muss durch einen Klick auf *Create Mask* die Maske erstellt werden. Um zu verhindern, dass ein falsches Echo verfolgt wird sollte mittels *Set Search Frame* der Bereich, in welchem die Biene das erste mal auftaucht, eingegrenzt werden (Das Suchfenster kann nur im Anzeigemodus *Tracking Spots* ausgewählt werden). Nun kann das Tracking durch den Button *Start* gestartet werden. über *Show Trackpoints* können die gefunden Trackpoints ausgegeben und gespeichert werden.

B.1 Hauptfenster

Funktionen der Schaltflächen des Hauptfensters (siehe Abbildung 14):

1 << Springt ein Bild zurück.

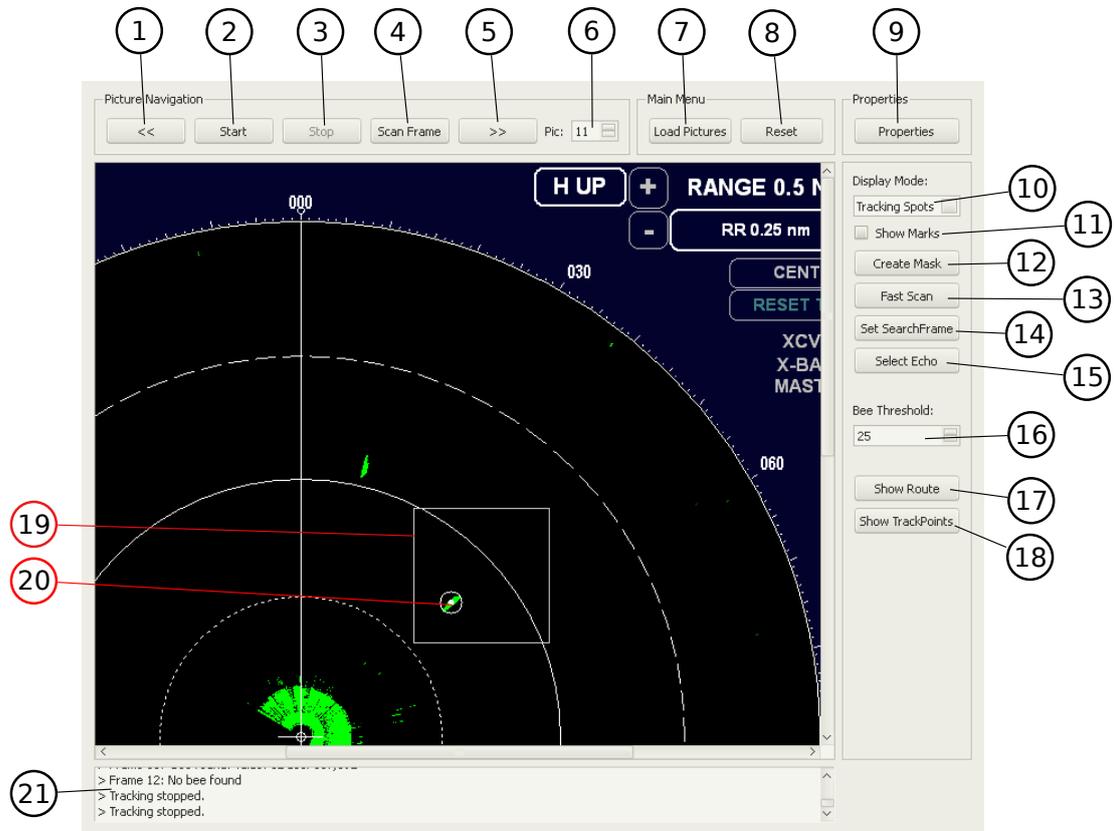
⁵Eclipse CDT: <http://www.eclipse.org/cdt/>

⁶OpenCV: <http://sourceforge.net/projects/opencvlibrary/>

⁷QT: <http://qt.nokia.com/products/qt/index.html>

2	Start	Startet den Tracking Vorgang mit den eingestellten Parametern.
3	Stop	Stoppt den Tracking Vorgang.
4	Scan Frame	Sucht im aktuellen Frame nach einem Echo.
5	>>	Springt ein Bild vorwärts.
6	Pic	Springt zu dem angegebenen Frame.
7	Load Pictures	Öffnet File-Dialog zum Laden von Radarbildern.
8	Reset	Setzt die Maske und bisher getrackten Punkte zurück.
9	Properties	Öffnet das Properties-Menü (siehe 15).
10	Display Mode	Ändert den Anzeige Modus: <i>Source Image</i> zeigt das Ausgangsbild an, <i>Mask</i> zeigt die Maske an, <i>Masked Image</i> zeigt das aktuelle maskierte Frame an, <i>Tracking Spots</i> zeigt das Suchfenster an und markiert gefundene Echos, <i>Show Nothing</i> beschleunigt das Tracken, ohne dass ein Bild angezeigt wird.
11	Show Marks	Zeigt Ortsmarken, die zuvor in der Datei resources/radarmarks.txt definiert wurden, im Radarbild an.
12	Create Mask	Erstellt die Maske aus 20 zufälligen Bildern.
13	Fast Scan	Erstellt die Maske, deaktiviert die Anzeige und startet das Tracking.
14	Set Search Frame	Definiert ein manuelles Suchfenster auf dem Radarbild.
15	Select Echo	Markiert einen Blob manuell auf dem Radarbild. Sucht den Schwerpunkt des ausgewählten Blobs.
16	Bee Threshold	Ändert den Bee Threshold (siehe Tracking Parameter).
17	Show Route	Zeigt die bisherige Trajektorie als Graph an. Das Bild kann gespeichert werden.
18	Show Track-Points	Zeigt eine Liste der getrackten Punkte an und filtert diese (siehe Abschnitt 6. Liste lässt sich abspeichern.
19	<i>Suchfenster</i>	Suchfenster-Markierung
20	<i>Echo</i>	Markierung des getrackten Echos
21	<i>Textfeld</i>	Ausgabe von Statusmeldungen und Informationen

Abbildung 14: Hauptfenster des Programms



B.2 Tracking Parameter

Alle Tracking Parameter sind über das *Properties* Menü (siehe Abbildung 15) einstellbar:

Radar Origin:	Die Position des Radargeräts auf den Radarbildern. [pixel] Find Origin findet diese Position automatisch.
Timesteps btw Frames:	Der Zeitintervall zwischen 2 Frames. [1/10s]
Background Color:	Die Hintergrundfarbe des Radarschirms. [RGB]
Echo Color:	Die Farbe der Radar-Echos. [RGB]
Max Bee Speed:	Die maximale Geschwindigkeit der Biene (Parameter zur Vergrößerung des Suchfensters). [m/s]
Max Search Frame Size:	Die maximale Größe des Suchfensters. [pixel]
Bee Threshold:	Schwellwert ab welcher Größe ein Blob als Echo erkannt wird.
Gaussian Filter Size:	Die Größe des Gaußfilter Kerns. Beeinflusst den Bee Threshold Wert. [pixel ²]
Resolution:	Die Auflösung der Bilder (Welche Entfernung repräsentiert ein Pixel auf dem Radarschirm). [m/pixel]
Mask Threshold:	Dieser Wert bestimmt, zu welchem Anteil ein Pixel in den Bildern vorkommen muss, um maskiert zu werden.
Point Blank Radius:	Maskiert einen Kreisbereich um das Radargerät um starke Reflektion im Nahbereich zu maskieren. Zum Deaktivieren auf 0 setzen. [pixel]

Abbildung 15: Properties Dialog

The screenshot shows a 'Properties Dialog' window with the following fields and controls:

- Radar origin (x,y):** Two input boxes containing '504' and '512', and a 'Find Origin' button.
- Timesteps btw frames:** An input box containing '10', with the unit 'in 1/10s' to its right.
- Background color:** Three color selection boxes (each with a small square icon) containing '0', '0', and '0', with the label 'RGB color' to the right.
- Echo color:** Three color selection boxes containing '2', '255', and '2', with the label 'RGB color' to the right.
- Max bee speed:** An input box containing '40', with the unit 'in pixel/s' to its right.
- Max Search Frame Size:** An input box containing '120', with the unit 'in pixel' to its right.
- Bee Threshold:** An input box containing '25', with the text 'smaller thresholds detect smaller echos' to its right.
- Gaussian Filter Size:** An input box containing '25', with the text 'sets the size of the filter kernel' to its right.
- Resolution:** An input box containing '4.026', with the unit 'in meter/pixel' to its right.
- Mask Threshold:** An input box containing '0.3', with the text 'btw [0.0,1.0]' to its right.
- Point Blank Radius:** An input box containing '58', with the text 'masked circle around the origin' to its right.

At the bottom of the dialog are three buttons: 'Save', 'Cancel', and 'Default'.

Literatur

[K. von Frisch, M. Landauer 1993] Karl von Frisch, erg. v. Martin Landauer (1993), *Aus dem Leben der Bienen*

- [R. Menzel 2009] Randolph Menzel (2009) , *Landschaftsgedächtnis der Bienen, aus Biologie in unserer Zeit p31-40*
- [N. Fortin 2008] Norbert Fortin (2008), *Navigation and Episodic-Like Memory in Mammals*
- [L. F. Jacobs, F. Schenk 2003] Lucia F. Jacobs, Françoise Schenk (2003), *Unpacking the Cognitive Map*
- [B. Fischer, W. Haaß 2007] Bernd Fischer, Wolf-Dieter Haaß (2007), *Radar Transponder für das Projekt Navigation und Tanzkommunikation von Bienen*
- [O’Neal, Landis, Rothwell, Kempel, Reinhard 2005] O’Neal, M.E., D. A. Landis, E. Rothwell, L. Kempel, and D. Reinhard (2005), *Tracking Insects with Harmonic Radar: a Case-Study with Carabids in Annual Crop Habitat. The American Entomologist. 50: 212-218.*
- [J. Riley, A. Smith 2002] J.R. Riley, A.D. Smith (2002), *Design considerations for an harmonic radar to investigate the flight of insects at low altitude*
- [Gady Agam 2006] Gady Agam (2006), *Introduction to programming with OpenCV*, <http://www.cs.iit.edu/~agam/cs512/lect-notes/opencv-intro/opencv-intro.html>
- [Qt Reference] Nokia Cooperation (2008), *QT Reference Documentation*, <http://doc.trolltech.com/>