

1 Modellentwurf und Simulation

Nach der sehr detaillierten Analyse der Messwerte wird in diesem Kapitel ein Modell aus den aufbereiteten Daten gewonnen und dieses anschließend in eine Simulationsumgebung implementiert. Aus den Messungen muss zunächst qualitativ eine Aussage über das Verhalten der BLE-Signale getroffen werden, um nachfolgend ein geeignetes mathematisches Modell zu finden. Dieses gewählte Modell wird darauf an die Gegebenheiten mit einer Parameterschätzung angepasst. Für die Anpassung wird zunächst ein Minimierungsproblem anhand der Messwerte aus dem vorigen Kapitel erstellt, welches mit einem Partikel-Schwarm-Optimierer gelöst wird. Die daraus gewonnenen Parameter lassen sich in die Modellgleichung einsetzen, um mit dessen Hilfe die Signalausbreitung der Beacons in einem Gebäude vorherzusagen.

Die Vorhersage der Ausbreitung von BLE-Funkwellen in einer Simulation kann dazu genutzt werden, mithilfe von einem Optimierungsverfahren die Leuchtfeuer in einem Raum so anzubringen, dass ein Lokalisierungssystem aus dieser Technologie mit maximaler Genauigkeit konfiguriert werden kann. Um zuerst die Abmessungen des betreffenden Raumes zu ermitteln, wird der Grundriss durch die Verwendung des Scitos G5 und der Software Miracenter als Bilddatei erstellt. Somit gelten die Wände als Begrenzung und der verbleibende Rest als Platz für die Signalausbreitung. Die maximale Genauigkeit der Positionsbestimmung würde logischerweise erreicht werden, wenn der komplette Raum mit Beacons unter Einhaltung des Mindestabstandes übersät wäre, was jedoch nicht ökonomisch ist. Deswegen wird zusätzlich die Anzahl der Beacons in die Problemformulierung mit eingearbeitet, sodass für jede mögliche Beacon-Anzahl eine optimale Konfiguration existiert.

1.1 Modellbildung

Zunächst wird in diesem Abschnitt das zugrunde liegende Modell für das Verhalten der Signalausbreitung der Beacons bestimmt. Dazu wird aus Tabelle 1.1 ein geeignetes Kanalmodell ausgesucht und dessen ungefährer Verlauf bestimmt. An diesem Verlauf wird es an den realen Messwerten validiert und anschließend in dessen Qualität durch die Parameterschätzung verbessert. Die genannten Modelle in der Tabelle können mit ihren Spezifikationen im Grunde alle als Ausbreitungsmodell für die Leuchtfeuer dienen. Jedoch fällt die erste Wahl auf das sogenannte WINNER II Modell, weil in dessen

Beschreibung die Anwendung in großen Räume mit hohem Aufkommen von Datenverkehr steht. Und da eben diese Gegebenheit in den Messungen aus dem vorigen Kapitel vorhanden waren und für den späteren Einsatzorte (z.B. Kaufhäuser, Flugplätze etc.) der Beacon-Systeme gegeben sein werden, bietet sich dieses Modell besonders vor den anderen an. Zudem beträgt die Reichweite eines Beacons nur 70 Meter, wohingegen die

Modell	Antennenhöhen	Frequenz	Bereich	In-/Outdoor	Bemerkungen
Free Space Loss	nicht begrenzt	$f \geq 30$ MHz	LOS begrenzt	Indoor-Indoor- bzw. Outdoor-Outdoor-Übertragungen bei direkter Sichtverbindung (LOS)	Systeme, bei denen direkte Sichtverbindung angenommen werden kann
Extended Hata	$h_1 = [1 \text{ m} ; 10 \text{ m}]$ $h_2 = [30 \text{ m} ; 100 \text{ m}]$	30 MHz – 3GHz	$d \leq 40$ km	automatisches Hinzuaddieren von Verlusten an Wänden bei Indoor-Outdoor- bzw. Indoor-Indoor Übertragungen (mehrere Räume/Gebäude)	mobile Dienste und andere Dienste, ohne direkte Sichtverbindung, nur im Bereich 2 – 3 GHz implementiert
Extended Hata-SRD	$h = [1,5 \text{ m} ; 3 \text{ m}]$	30 MHz – 3GHz	$d \leq 0,3$ km	automatisches Hinzuaddieren von Verlusten an Wänden bei Indoor-Outdoor- bzw. Indoor-Indoor Übertragungen (mehrere Räume/Gebäude)	Kurzstreckenverbindungen, bei denen wenigstens LOS angenommen werden kann
WINNER II	$h \leq 6 \text{ m}$	2 GHz – 6GHz	$d = [5 \text{ m} ; 100 \text{ m}]$	Indoor-Indoor-Übertragungen bei LOS oder NLOS	Anwendung in großen offenen Räumen bei hohem Datenverkehr (z.B. Konferenzhalle, Fabrikgebäude)

Tabelle 1.1: Übersicht der Kanalmodelle im Indoor-Bereich, in Anlehnung an [?]]

Geltungsbereiche der anderen Modelle dafür überdimensioniert erscheinen.

1.1.1 Das WINNER II Modell

Das WINNER II Modell [?], benannt nach dem „Wireless-World-Initiative-New-Radio“ (WINNER)-Konsortium, welches aus dem Zusammenschluss mehrerer Unternehmen besteht und die Verbesserung der Leistung von Mobilfunkkommunikationssystemen anstrebt, beinhaltet verschiedene Aspekte in der Berechnung von Ausbreitungsverlusten. Die Formel die dem Modell zu Grunde liegt, beschreibt den Verlust einer Signalsstärke in Abhängigkeit zu einem zurückgelegten Weg folgendermaßen:

$$L_{WINNER} [dBm] = A \cdot \log_{10}(d [m]) + B + C \cdot \log_{10}\left(\frac{f_c [GHz]}{5}\right)$$

In der Formel steht L_{WINNER} für den Leistungspegel in Dezibel Milliwatt (dBm), der das logarithmische Verhältnis aus einer Leistung in Form der Empfangsstärke im Vergleich zur anfänglichen Sendeleistung des Beacons angibt. So kann die obige Gleichung umgeschrieben werden zu:

$$10 \cdot \log_{10}\left(\frac{P_{Sendung} [mW]}{P_{Empfang} [mW]}\right) = A \cdot \log_{10}(d [m]) + B + C \cdot \log_{10}\left(\frac{f_c [GHz]}{5}\right)$$

oder auch direkt als dBm-Einheit formuliert werden:

$$L_{Sendung} [dBm] - L_{Empfang} [dBm] = A \cdot \log_{10}(d [m]) + B + C \cdot \log_{10}\left(\frac{f_c [GHz]}{5}\right) \quad (1.1)$$

Die Parameter der Gleichungen bestehen primär aus einem Koeffizienten A , der die in der Luft zurückgelegte Strecke des Signals proportional gewichtet. Desweiteren fließen zum einen noch der additive Einfluss von dem Parameter B für sonstige Störfaktoren und zum anderen der ebenfalls multiplikative Faktor C zur Gewichtung des Ausbreitungsverlustes durch die Frequenz f_c mit ein. In Abbildung 1.1 ist der qualitative Verlauf der Empfangsstärke, die nach oben hin abnimmt und der Distanz, welche nach rechts hin zunimmt, skizziert. Als Parameter wurden Werte aus der Fachliteratur [?] entnommen und in die Gleichung eingesetzt. Im Vergleich zu Abbildung ?? sieht der Funktionsverlauf weitgehend identisch aus, sodass nun mit der Parameterschätzung begonnen werden kann.

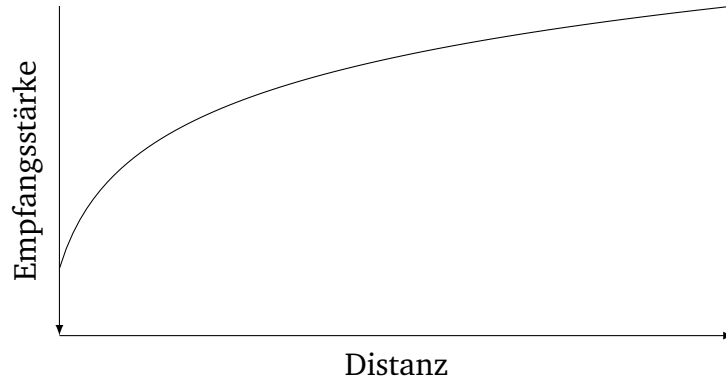


Abb. 1.1: Qualitativer Verlauf des WINNER II-Modells

1.1.2 Parameterschätzung

Um die qualitative Repräsentation des WINNER II-Modells in eine quantitative für die Beschreibung des Ausbreitungsverlustes von BLE-Signalen zu überführen, müssen die Parameter A , B und C dementsprechend angepasst werden. Die Gleichung aus 1.1 vereinfacht dabei sich nochmals, wenn f_c als konstant angenommen wird und die Ausdrücke aus B und C miteinander zu dem neuen Parameter B^* zusammengefasst werden. Die Anpassung wird dabei in Form einer Optimierungsaufgabe vorgenommen, wobei es gilt die richtige Distanz anhand der RSSI-Werte zu schätzen. Im vorangegangenen Kapitel ergaben die Messungen, dass unter Betrachtung aller Daten es sehr schwierig ist eine genaue Vorhersage zu treffen. Deswegen wurden nur die stärksten 100 Signale einer Distanz gemittelt und diese Ergebnisse unter ?? vorgestellt. Die Durchschnittswerte wurden anschließend zusammen mit den realen Distanzen in gleicher Reihenfolge der Länge k in die zwei Vektoren $d_{real}(k)$ und $RSSI(k)$ gespeichert. Aus der Umstellung der Gleichung 1.1 nach d lässt sich aus dem empfangenen RSSI-Wert die geschätzte Entfernung bestimmen. Mit all diesen Informationen lässt sich ein Optimierungsproblem so formulieren, das eine zu minimierende Kostenfunktion J als Differenz aus der direkt gemessenen Distanz von Beacon zu Smartphone zu der berechneten Distanz aufgefasst und durch deren folgende Quadrierung so eine parabolische Funktion erstellt wird. Die Quadrierung bewirkt, dass der Wert der Kostenfunktion von allen Seiten in Richtung des globalen Minimums abnimmt und somit das Optimierungsproblem einfacher zu lösen ist.

$$\min_{A, B^*} J = \sum_{k=1}^{k_{end}} \left(d_{real}(k) - 10^{\frac{L_{Sendung} - RSSI(k) - B^*}{A}} \right)^2 \quad (1.2)$$

Als Lösungsalgorithmus für die Optimierung wird ein Partikel-Schwarm-Algorithmus (PSO) gewählt. Der Partikel-Schwarm-Optimierer ist ein Algorithmus mit einer künst-

lichen Intelligenz, die es zur Aufgabe hat eine Funktion zu minimieren bzw. zu maximieren. Es ist dabei einfacher sich die Partikel als einen Vogelschwarm vorzustellen. Der Grundgedanke dabei ist, dass mehreren Partikel ein zufälliger Startwert $p(t)$ im Suchraum unter Berücksichtigung der Gleichungsnebenbedingungen zugeordnet wird. D.h. im übertragenden Sinn, dass ein Vogelschwarm auf einer Wiese ausgesetzt wird. Ausgehend von diesem Startwert werden die einzelnen Vögel/Partikel im ersten Schritt in verschiedene Richtungen mit unterschiedlichen Geschwindigkeiten geschickt. Anschließend werden an den Zielpunkten die Kostenfunktionen mit den Positionen (oder auch Parametersätzen) berechnet und so der Funktionswert bestimmt. Die Schwarmin-telligenz wird hierbei so realisiert, dass jedes Individuum seine aktuelle Position mit den zugehörigen Kosten kennt, sowie über die Position des Partikels mit den geringsten Kosten im ganzen Schwarm informiert wird. In Folge des Wissens um die beste Position, fliegen alle Vögel/Partikel in Richtung des lokalen Minimas und decken somit den gesamten Raum ab.

Als Gleichung aufgefasst wird die Geschwindigkeit v im nächsten Zeitschritt $t + 1$ eines Partikels aus der alten Geschwindigkeit $v(t)$, aus den drei konstanten Gewichtungen w , r_1 und r_2 , zwei Zufallszahlen c_1 und c_2 und verschiedenen Positionen berechnet. Die Positionen sind zum einen die aktuelle Position $x(t)$, die mit den besten Kosten des einzelnden Individuums p und die Position der geringsten Kosten im ganzen Schwarm g . Im ganzen lauten die Gleichungen zur Bestimmung des nächsten Punktes [?]:

$$v(t+1) = (w \cdot v(t)) + (c_1 \cdot r_1 \cdot (p - x(t))) + (c_2 \cdot r_2 \cdot (g - x(t))) \quad (1.3)$$

$$x(t+1) = x(t) + v(t+1) \quad (1.4)$$

Das Abbruchkriterium wird hierbei mit einer maximalen Anzahl an Generationen/Durchläufen der Hauptschleife angegeben. Der Vorteil dieses Verfahrens ist es, dass der komplette Suchraum abgetastet wird und durch die Intelligenz des Schwarms die Parametersätze mit den geringsten Kosten identifiziert werden. Der Nachteil ist der relativ hohe Rechenaufwand im Gegensatz zu anderen Verfahren (z.B. Newton, Trust-Region, Simplex etc.), zumal kein besseres Ergebnis mit dem PSO für ein solches Problem zu erwarten wäre. Jedoch baut der Erweiterte Partikel-Schwarm-Optimierer in einem der nächsten Abschnitte auf diesem Verfahren auf, sodass in diesem Teil der Arbeit die Vorstellung der einfachen Variante zur Einführung vorteilig ist.

Als Ergebnis der Parameterschätzung wurden in Abbildung 1.2 zunächst die berechneten RSSI-Werte für die jeweiligen Distanzen im Vergleich zu den gemessenen

Mittelwerte aller drei Beacons gegenübergestellt. Wie schon in der Analyse der Messwerte kritisiert, entspricht das Verhalten des Ausbreitungsverlustes ab dem zehnten Meter nicht mehr der theoretischen Annahme. Während der Verlauf vom ersten Abschnitt sehr gut geschätzt werden kann, versagt aber das einfache WINNER II-Modell bei größeren Distnazen. Ähnlich verhält es sich bei der Schätzung der Entfernung, wie Abbildung 1.3 zeigt. Hier wurde das Modell lediglich nach d umgestellt und als Eingabe dienen die Mittelwerte aus 1.2. In Tabelle 1.2 sind die Parameter aus der Optimierung aufgeführt, welche in das Modell zur Berechnung der letzten Diagramme eingesetzt wurden.

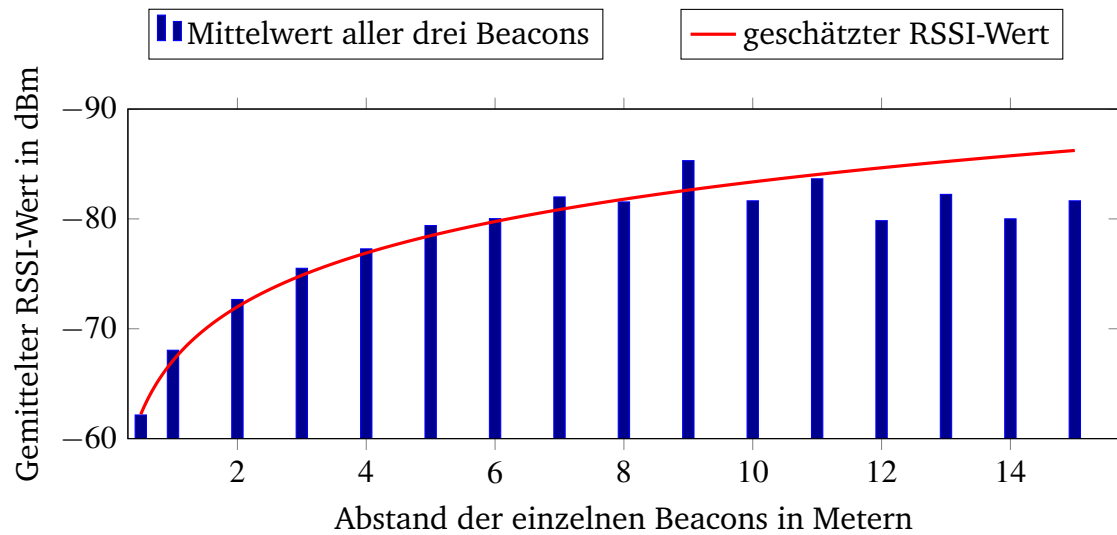


Abb. 1.2: Durschnittlicher RSSI-Werte im Vergleich zum berechneten RSSI-Wert aus dem WINNER II-Modell über die einzelnen Entfernungen

	Parameter	
	A	B^*
Wert	16,2623	55,1046

Tabelle 1.2: Parameter des Modells

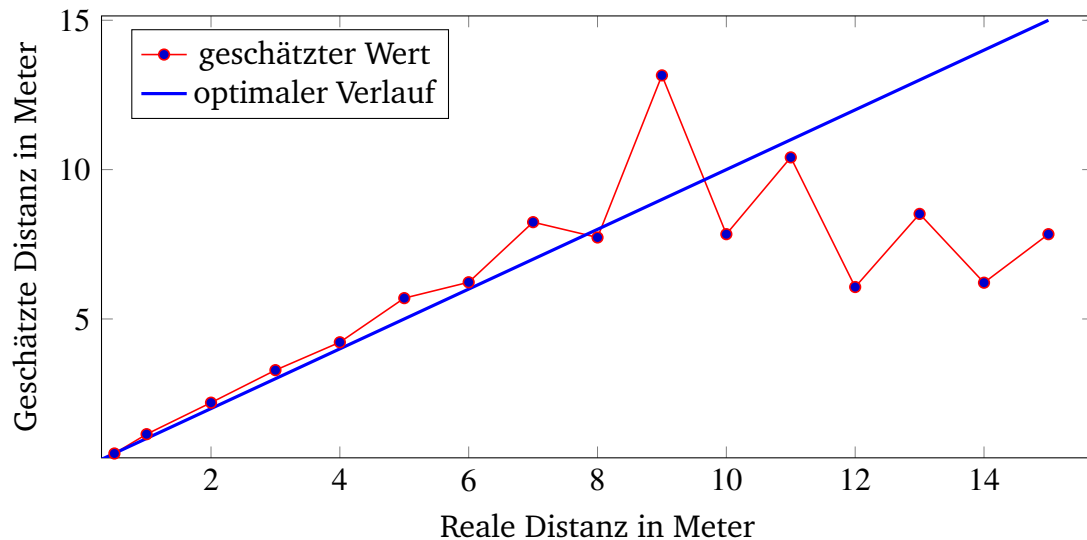


Abb. 1.3: Vergleich der Distanz-Schätzung mit dem WINNER II-Modell zur gemessenen Entfernung

1.2 Vertrauens-Skala

Im Hinblick auf die verschiedenen Betriebsmodi der Beacons, den unterschiedlichen Arten der Positionsbestimmung (Landmarkensystem und Mikro-Lokalisierung) und vielen noch nicht vorgestellten Techniken mit denen sich die Vorhersage der Position bestimmen und verbessern lassen, fällt es sehr schwer die Güte der Leuchtfeuer-Signale einzuschätzen. Mit Güte soll gemeint sein, inwieweit dem Modell 1.1 vertraut wird die aktuelle Position zu schätzen und wie genau diese Schätzung in Metern oder Zentimetern erfasst werden kann. Um unabhängig von der Art der Lokalisierung, dem Anwendungsbereich und den eingesetzten Techniken die Güte angeben zu können, wird lediglich die Wahrscheinlichkeit mit dem WINNER II-Modell die tatsächliche Distanz zu ermitteln, als Wert dafür angenommen. D.h. aus den Messungen der RSSI-Werte vom vorigen Kapitel werden diesmal alle Daten für die Distanzberechnung herangezogen und mit einer Toleranz versehen. Die berechneten Entfernungen werden darauf mit realen Distanzen verglichen und im Falle, dass sie sich nur in einem definierten Spielraum voneinander unterscheiden, wird dies als eine erfolgreiche Positionsbestimmung angesehen. Der prozentuale Anteil aller erfolgreichen Berechnungen wird so als Gütemaß gewertet und somit wird Qualität der Beacon-Signale als eine statistische Wahrscheinlichkeit, des Empfanges einer für das Modell geeigneten Signalsstärke pro Messung, aufgefasst. Die Ergebnisse aus diesen Überlegungen finden sich in Abbildung 1.4 und 1.5 wieder. Sie zeigen einmal die Wahrscheinlichkeit W_{berech} ein gültiges Signal zu empfangen bei einer Toleranz von 0,5 und bei 2 Metern, wobei die dazugehörigen festen Distanzen die

jeweiligen gemittelten drei Messreihen d_{mitt} darstellen. Zudem wurde versucht die Güte als Funktion zu charakterisieren. Dazu wurde die Funktion 1.5 aufgrund ihrer Art ausgewählt und um weitere Parameter erweitert 1.6, welche schließlich durch die Lösung des Minimierungsproblems 1.7 und unter der Verwendung vom PSO bestimmt wurden (Ergebnis siehe Tabelle 1.3).

$$W_{berech} = \frac{1}{d_{mitt} + 1} \quad (1.5)$$

$$W_{berech} = \frac{1}{p_1 \cdot d_{mitt} + p_2} + p_3 \quad (1.6)$$

$$\min_{p_1, p_2, p_3} J = \sum_{k=1}^{k=15} (W_{real}(k) - W_{berech})^2 \quad (1.7)$$

Der Vorteil dieser Betrachtweise liegt in den Funktionen der Beacons, da hier die Frequenz bzw. Häufigkeit der Sendevorgänge eingestellt werden kann. Durch die Erhöhung der Übertragungen pro Sekunde lässt sich somit die Zeit verringern, bis ein gültiges Signal das Messgerät erreicht. Anhand dieser Einstellmöglichkeiten lässt sich zusammen mit der hier vorgestellten Skala die Dynamik der Messungen beeinflussen.

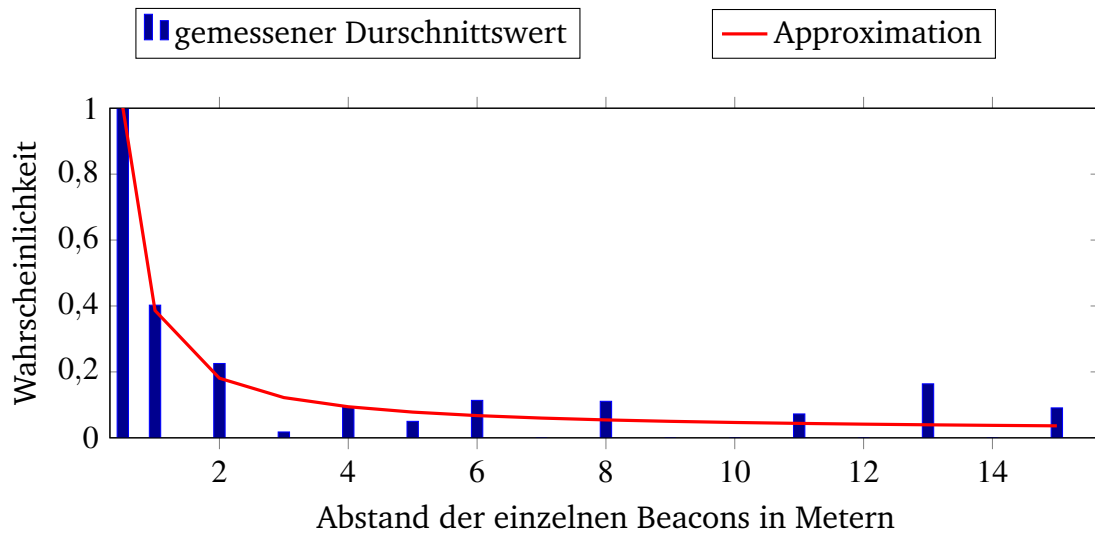


Abb. 1.4: Wahrscheinlichkeit des Empfangs gültiger Signale in einem Toleranzbereich von 0,5 Metern und dessen Approximation

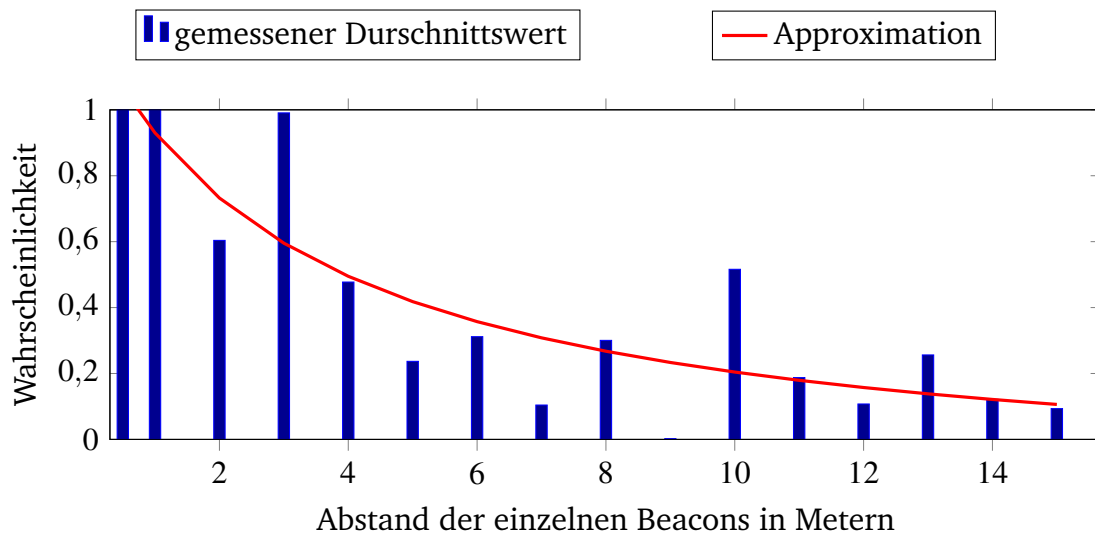


Abb. 1.5: Wahrscheinlichkeit des Empfangs gültiger Signale in einem Toleranzbereich von 2 Metern und dessen Approximation

		Parameter		
		p_1	p_2	p_3
Toleranz	0,5 m	3,3777	-0,6746	0.0163
	2 m	0,2035	0,7145	0,1594

Tabelle 1.3: Parameter der Vertrauensskala

1.3 Entwicklung der Simulationsumgebung

Mit den gewonnen Erkenntnissen aus der Modellbildung können nun Vorhersagen über die Signalausbreitung und der zu erwartenden Genauigkeit der Entfernungsschätzungen getroffen werden, die schließlich zu einer optimalen Planung eines Indoor-Lokalisierungssystems genutzt werden können. Dazu wird zunächst eine Entwicklungsumgebung zur Herstellung der Simulation benötigt. Als Basis der Simulation wird eine grafische Oberfläche (GUI) mit dem Editor GUIDE erstellt, der Teil des Programmes MATLAB [?] ist. In dieser Sektion werden die einzelnen Funktionen der GUI besprochen und deren Verarbeitungskette vorgestellt. Hierfür wird als erstes die Konvertierung der Zeichnung eines Grundrisses (als PNG-Bilddatei) in verarbeitbare Informationen erläutert, die von substantieller Bedeutung für den gesamten Simulations-Prozess ist. Anschließend wird in der weiteren Implementierung speziell der Aspekt einer optimalen Verteilung, unter besonderer Betrachtung der Vertrauens-Skala, aufgegriffen. Eigens dafür wurde der vorher vorgestellte Partikel-Schwarm-Algorithmus aus der Parameterschätzung erweitert, sodass er auch für den abstrakten Fall einer Anordnung

von Funkbaken eingesetzt werden kann. Abschließend wird mit einem fiktiven Grundriss das komplette Lighthouse Keeper-Prinzip exemplarisch vorgeführt.

1.3.1 Grundrissanalyse

Für eine weitgehende Automation der gesamten Planung müssen die Grundrisse bzw. Rauminformationen, die mit der Software Miracenter (siehe Abschnitt ??) generiert wurden, vor der Simulation ausgewertet werden. In der Auswertung oder Analyse dieser Daten muss das Programm selbstständig in der Lage sein, eine Wand oder generell einen Raum zu erkennen. Zudem müssen auch Befestigungspunkte definiert sein, an denen die Beacons angebracht werden können. Die Umsetzung dieser Funktionen basiert hierbei auf einer Eigenentwicklung, da bisher keine vergleichbaren Arbeiten existieren. Für die Erkennung von Wänden respektiven Räumen könnten zwar Algorithmen eingesetzt werden, die beispielsweise Formen erkennen oder Cluster von Farbgruppierungen bilden. Im Hinblick jedoch auf mögliche Erweiterungen der gesamten Problematik, z.B. mit Hindernissen oder mit Bereichen, in denen keine Beacons befestigt werden können, bietet sich das hier selbstentwickelte Verfahren an. Für ein besseres Verständnis wird der allgemeine Vorgang diesbezüglich in Abbildung 1.6 illustriert. Das Beispiel zeigt einen quadratischen Raum, wobei die Wände schwarz und die freien Räume in weiß gehalten sind. Die einzelnen Unterteilungen sollen die Pixel darstellen, sodass eine Auflösung von 7×7 Pixeln gegeben ist. In einer ersten Phase werden in vertikaler Reihenfolge die einzelnen Pixel von „links nach rechts“, sowie deren Vorgängerpixel und deren Nachfolgerpixel betrachtet. Im vorliegenden Fall werden dafür zwei Matrizen mit 7×7 Einträgen initialisiert, in der jeder Eintrag zunächst auf Null gesetzt ist. Wechselt in der abgefragten Reihe ein weißer Pixel auf einen schwarzen, so wird die Position des schwarzen Pixels in der Befestigungs-Matrix um eins erhöht und in der Raum-Matrix wird die Position des weißen Pixels ebenfalls mit eins addiert. Die folgenden weißen Pixel werden anschließend auch als Raumkoordinaten markiert, solange kein weiterer schwarzer Punkt erscheint. Falls eine weitere Wand im Verlauf der Reihe detektiert wird, setzt die Raummarkierung aus und lediglich der schwarze Bildpunkt wird als Befestigung eingeschrieben.

Diese Vorgehensweise alleine ist jedoch nicht robust genug, was auch in der Abbildung skizziert wurde. In diesem Fall wurden zwei Befestigungen und zwei Raumpunkte falsch interpretiert, weil der Algorithmus nur den unmittelbaren Nachbarn eines Bildpunktes betrachtet und somit nicht feststellen konnte, ob der Raum abgeschlossen ist oder nicht. In Folge dieses bekannten Fehlers wurde für alle weißen Pixelkoordinaten

an den Bild-Rändern eine rückwärtsgerichtete Suche implementiert, die bis zur nächsten Wand alle Einträge beider Matrizen um eins reduziert, wie auch rechts neben der ersten Phase zu sehen ist. Somit wird sichergestellt, dass die Räume stets abgeschlossen sind und keine Befestigungen an den Ecken existieren. Die gleiche Vorgehensweise wird dreimal in jede Richtung („von rechts nach links“, „von unten nach oben“ und „von oben nach unten“) wiederholt. Dabei entstehen Vertrauenswerte in den Matrizen, anhand derer über die endgültige Definition nach Befestigungen und Räumen entschieden wird. In den verschiedenen Versuchen und Testszenarien mit offenen Türen und verwinkelten Gängen erwies sich der vorgestellte Algorithmus als sehr zuverlässig. Zusätzlich wurde die Analyse auf den Rot-, Grün- und Blau-Farbraum (RGB) erweitert, sodass mit weiteren Farben spezielle Objekte erfasst werden. Z.B. können nun mit roten Farben Bereiche gekennzeichnet werden, an den keine Beacons befestigt werden können.

Damit die Bildinformationen in den metrischen Raum abgebildet werden können, muss zunächst deren Verhältnis von Pixel zu Meter bekannt sein. In der Simulation kann dieses Verhältnis direkt angegeben werden, indem die Abmessungen des gesamten Raumes in die GUI eingetragen werden. Aus diesen Informationen berechnet die Software eigenständig die Verhältnisse und überträgt die Pixel in metrische Koordinaten. Anschließend werden die Wände bzw. Befestigungen durch einfache Polygone ersetzt, wobei die Ecken von Wänden als Referenzpunkte genommen werden. Dies hat den Vorteil, dass für Aufteilung der Beacons entlang dieser Linien nur Abstandswerte kalkuliert werden müssen. Zudem kann die Kollision von Signalen und Wänden schneller berechnet und somit entschieden werden, ob sich ein Beacon in der LOS oder NLOS zu einem Punkt im Raum befindet. Im letzten Schritt der Analyse wird der Raum in einzelne Elemente untergliedert. Die Größe der Quadrate wird dazu in die Oberfläche der Simulation eingetragen. Die Aufteilung in finite Elemente ergibt sich aus der Nutzung der Vertrauens-Skala, die abhängig von der gewünschten Toleranz bzw. Lokalisierungs-Genauigkeit ist. Diese Toleranz bildet die Größe eines Raumelementes, in der die modellbasierte Entfernungsschätzung (\pm der Toleranz) mit einer bestimmten Wahrscheinlichkeit den tatsächlichen Abstand von Raumelement zu Beacon angibt.

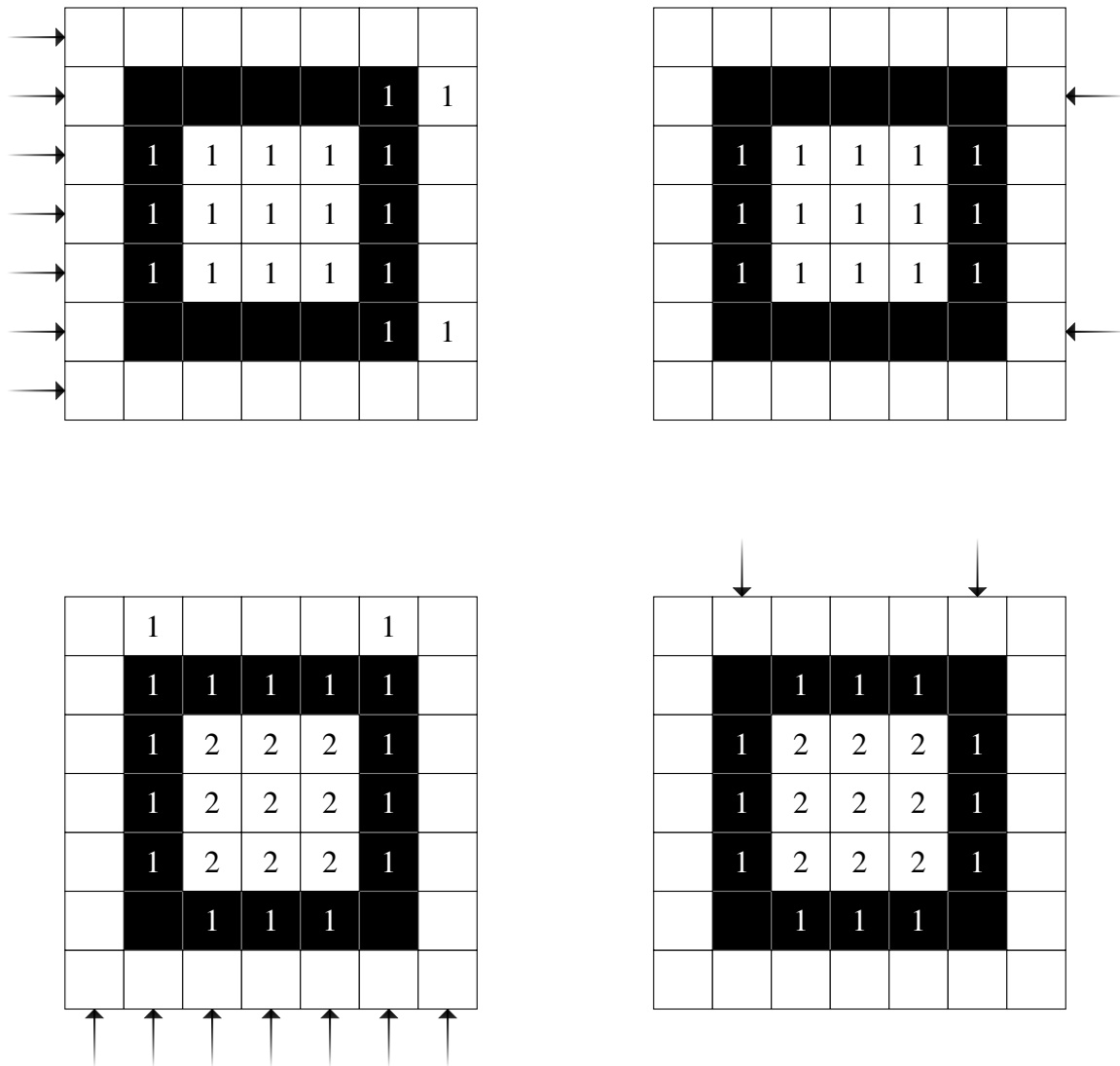


Abb. 1.6: Schematischer Ablauf der Grundriss-Analyse

1.3.2 Optimierung der Beacon-Infrastruktur

Das erklärte Ziel des Lighthouse Kepper-Konzeptes ist es, eine größtmögliche Genauigkeit in der Lokalisierung von Objekten im Indoor-Bereich, bei einer minimalen Anzahl von Beacons zu erreichen. Der Unterschied zu den bisherigen Optimierungsverfahren in dieser Arbeit besteht zum einen in den zu beeinflussenden Parametern und zum anderen in der mehrkriteriellen Optimierungsaufgabe. Während bisherige Parameter zu den rationalen Zahlen gehörten und sie somit kontinuierlich änderbar waren, existieren in dieser Optimierungsaufgabe lediglich Parameter als Boolesche Variablen (Beacon platziert/nicht platziert). Zudem muss die Kostenfunktion nach zwei Kriterien (Genauigkeit und Anzahl) optimiert werden, weswegen ein Ansatz in Richtung Pareto-Optimierung führt.

Im folgenden Abschnitt wird der Algorithmus zur Optimierung der Beacon-Infrastruktur näher erläutert. Die Optimierung findet dabei unter Zuhilfenahme der Vertrauensskala als Genauigkeitsmaßstab und der Verwendung des Modells zu den Ausbreitungsverlusten von BLE-Signalen statt. Im zweiten Teil wird auf die Bewertung der beiden unterschiedlichen Kriterien eingegangen und dazu wird der Begriff Pareto-Front näher erklärt. Ein Beispiel für die Optimierung findet sich im Anschluss an diese Sektion in dem Beispiel für den Lighthouse Kepper.

Erweiterter Partikel-Schwarm-Algorithmus

Derweil genetische Algorithmen in den Bereichen der mathematischen Problemformulierungen nur eine geringe Rolle spielen und nur in wenigen Ausnahmen bessere Ergebnisse als herkömmliche Verfahren liefern, stellen sie bei Problemen mit Entscheidungszweigen die einzige praktikable Lösung dar. Die Arbeit hinter der Erweiterung des PSO wurde dabei maßgeblich noch von zwei weiteren genetischen Algorithmen beeinflusst. Es finden sich Teile des „Strength Pareto Evolutionary Algorithm 2“ (SPEA2) [?] sowie des „Simulated Annealing“-Verfahrens [?] in dieser neuen Methode wieder. Denn erst die Kombination aller genannten Ansätze lieferte die gewünschten Ergebnisse. Für die eigentliche Optimierung werden zunächst die Kostenfunktionen beider Merkmale benötigt. Indes die Funktion der Anzahl der Beacons mit $\min J_{Anzahl}$ trivial erscheint, unterliegt die Lokalisierungsgenauigkeit mehreren Faktoren, die je nach Anwendungsfall unterschiedlich gewichtet werden können. Statt unterschiedlichen Formulierungen des Problems wird stellvertretend für die Qualität von Signalen die Vertrauensskala als allgemeine Referenz gewählt. Es wird davon ausgegangen, dass die Ortungs-Genauigkeit beispielsweise mithilfe der Trilateration überall gleich ist, solange Beacon-Signale empfangen werden. Es hängt jedoch von den Wahrscheinlichkeiten ab, wie lange auf ein gültiges Signal gewartet werden muss, bis eine Positionsbestimmung die gewünschten Ergebnisse liefert.

Ausgehend von der Forderung eine bestmögliche Abdeckung des Raumes mit Beacon-Signalen zu erreichen, stellt sich die Frage nach der Relation der Vertrauenswerte in jedem Abschnitt des Raumes. Würden angenommen zwei Beacons relativ nah nebeneinander liegen, würde in ihrem unmittelbaren Umkreis der Vertrauenswert besonders hoch liegen, falls von beiden Beacons die Werte addiert werden. Jedoch sind zwei Beacons in einem konzentrierten Bereich wenig effektiv, wenn in einer entfernten Ecke eines Raumes dafür keine Beacons platziert wurden. Unter Betrachtung dieser Überle-

gung wurde so eine Gütekriterium festgelegt, die sich aus dem höchsten Vertrauenswert eines Beacons in jedem Raumelemente zusammensetzt. Mathematisch formuliert werden über die Anzahl aller Beacons n und über die Menge der Raumelemente m zuerst die einzelnen Entfernungen $d_{n,m}$ bestimmt. Darauf folgend wird anhand der Gleichung 1.6 die einzelnen Vertrauenswerte $W_{n,m}$ geschätzt und daraus das Gütemaß mithilfe der Summe der besten Signale in einem Raumelement $best \{W_{1...n_{end},m}\}$ gebildet. Somit ergibt sich folgendes Maximierungsproblem:

$$d_{n,m} = \sqrt{(Pos_{n,x}^{Beacon} - Pos_{m,x}^{Element})^2 + (Pos_{n,y}^{Beacon} - Pos_{m,y}^{Element})^2}$$

$$W_{n,m} = \frac{1}{p_1 \cdot d_{n,m} + p_2} + p_3$$

$$\max_{Pos^{Beacon}} J_{Qualität} = \sum_{m=1}^{m_{end}} best \{W_{1...n_{end},m}\}$$

Der erweiterter Partikel-Schwarm-Algorithmus muss nun nach den Positionen kleinen Sender optimieren, die schon in der Grundrissanalyse vorgegeben wurden. Dabei bedient sich das Verfahren an dem Belohnungs-/Bestarfungssystem (oder auch Selektion genannt) aus SPEA2 und der Heuristiken aus dem Simulated Annealing. In der Initialisierungsphase werden die einzelnen Populationen des Schwarms mit Beacons voll besetzt, sodass diesmal beim PSO jeder Partikel den gleichen Startwert besitzt. Lediglich im Belohnungs-/Bestarfungssystem werden die Gewichtungen anfangs für eine Zunahme bzw. Abnahme von Beacons zufällig zwischen den Werten 0 und 1 vergeben. Im nächsten Schritt wird für jede Konfiguration die jeweilige Güte J berechnet und anschließend im Bewertungssystem mit der vorigen Generation verglichen. Als Maßstab für eine bessere oder schlechtere Generation kommen daher die vorher definierten Kriterien zum Einsatz. Dabei werden im Endeffekt die Konfigurationen mit der geringen Anzahl an Beacons faktisch besser bewertet. Dieser Schritt war notwendig, da mit weniger Beacons die räumliche Abdeckung naturgemäß abnimmt und somit $J_{Qualität}$ nie ansteigen wird. Jedoch lassen sich Partikel mit gleicher Beacon-Anzahl in ihrer Qualität vergleichen und hier wird das Individuum belohnt, welches die höchste Güte in der Vertrauensskala aufweist und das mit dem niedrigerem Wert bestraft. Dementsprechend werden die jeweiligen Gewichtungen in den Grenzen 0 und 1 prozentual erhöht bzw. gesenkt. Im Folgenden kommt der Charakter des PSO zum Einsatz. Denn bei der Neuordnung der Konfigurationen werden alle Gewichtungen eines Partikels mit speziellen Individuen verrechnet. Diese Partikel bzw. deren Gewichtungen bestehen aus denen mit der besten Fitness im Schwarm und aus der eigenen besten Fitness, die jemals erreicht wurde. Im Grunde werden die einzelnen Gewichtungen zur Wegnahme oder Anbringung eines Beacons einander gegenüber gestellt, wobei sich der größere

Wert durchsetzt. Tritt dabei nach einer unbestimmten Iterationszeit ein totes Ende auf, d.h. das alle Gewichtungen genau 0 oder 1 sind, behilft sich der Algorithmus mit den Heuristiken des Simulated Annealing-Verfahrens. In diesem speziellen Fall wird ein totes Ende detektiert und das betreffende Individuum „mischt“ seine Gewichtungen mit anderen zufällig gewählten Partikeln und findet meist so aus einem lokalen Optimum heraus, solange es sich nicht im globalen befindet.

Zur Veranschaulichung wird hier noch einmal die Hauptschleife im ersten Aufruf durchlaufen:

Eingabe: **N** (Größe des Schwarms)
D (Dimension des Problems – Anzahl an Befestigungspunkten)
I (Maximale Anzahl an Generationen)
KR (Mittelpunkt-Koordinaten der Raumelemente)
KB (Koordinaten der Befestigungen)
Ausgabe: **F** (Fitnesswerte einer jeden Generation)
E (Konfigurationen mit den besten pareto-optimierten Kriterien)

Schritt 1: **Initialisierung:** Generierung der Konfigurationen \mathbf{B}_0 und aller Gewichtungen $\mathbf{G}_{Zu,n}$ und $\mathbf{G}_{Weg,n}$ der Beacons für alle Partikel.

Schritt 2: **Fitnessbestimmung:** Berechnung der Fitnesswerte für jedes Individuum im Schwarm.

Schritt 3: **Bewertung:** Vergleich aller Kopien der vorherigen Generation mit der neuen. Falls die neuen Parametersätze optimalere Ergebnisse liefern, werden alle Zustände auf Unterschiede untersucht und lediglich die Veränderungen belohnt. Bei schlechteren Fitnesswerten werden sie hingegen bestraft.

Schritt 4: **Schwarmintelligenz:** Mit den Gewichtungen der besten Konfigurationen und der aktuellen Generation werden die Beacons neu verteilt.

Schritt 5: **Totes Ende-Detektierung:** Falls keine Änderungen mehr eintreten, werden die Eigenschaften der Partikel zufällig mit denen anderer Individuen auf der Pareto-Front kombiniert.

Paretofront

Mehrkriterielle Probleme werden häufig als Pareto-Optimierung, nach dem italienischen Ingenieur, Ökonom und Soziologe *Vilfredo Federico Pareto*, bezeichnet. Die Optimierung verläuft dabei unter Betrachtung mehrerer Kostenfunktionen, wobei die Veränderungen der Kosten eines Problems sich auf die der anderen auswirkt. Jedoch bedingt die Verbesserung eines Kriteriums in der beacon-Problematik die qualitative Verschlechterung des anderen. Wie weit diese Wechselwirkungen ausgeprägt sind, zeichnet sich beim Übertragen der Fitnesswerte in den gleichen mathematischen Raum ab. Dabei bildet sich eine Front, in dessen Richtung des negativen Gradienten keine weiteren Lösungen zu finden sind. Der Vorteil der in Abbildung 1.7 dargestellten Pareto-Front ist die Möglichkeit zwischen den Kriterien abzuwägen. Falls der Kunde eine qualitativ hochwertige und flächendeckende Lokalisierung benötigt, kann er sich aus der Menge an Lösungen eine für ihn ökonomische optimale Konfiguration auswählen. Für den Kunden mit geringeren Ansprüchen würden Konfigurationen mit weniger Beacons ausreichen, welche er sich unter den gleichen Optimalitätsbedingungen auswählen kann und somit ist die Anforderungen an eine universelle optimale Planung erfüllt.

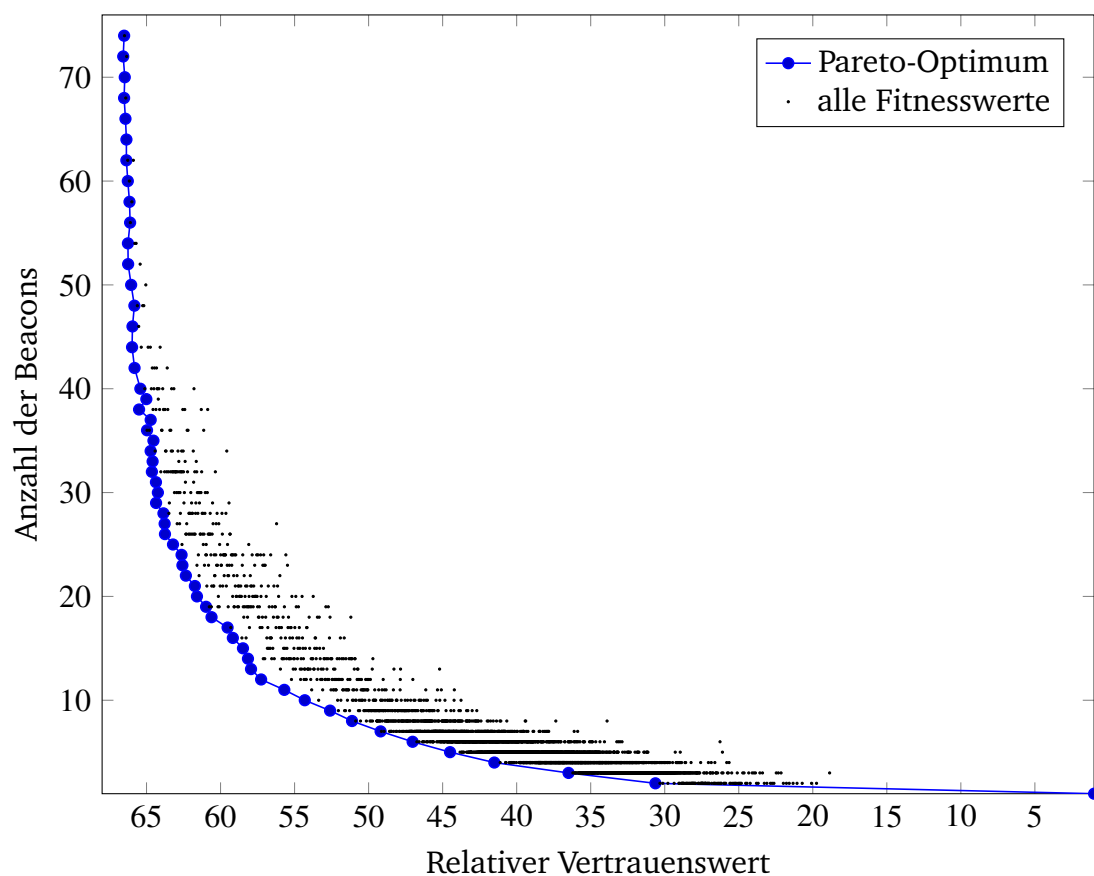


Abb. 1.7: Beispiel einer Paretofront

1.3.3 Simulationsbeispiel

Zum Abschluss der Thematik wurde in Abbildung 1.8 einmal eine komplette Simulation durchgeführt. Als test diente dazu ein fiktiver Raum in quadratischer Form mit den Abmessungen von $11,9 \times 7$ Metern. Zur Vereinfachung wurde die gleiche Höhe für die Sender wie für die Empfänger gewählt, welche denen in den statischen Experimenten gleichen. Als Toleranz für die Vertrauenskala wurde der Wert von einem Meter gewählt. Diese Wert wurde gleichsam in die Größe der Raumelemente übernommen, die schachbrettförmig über die gesamte Grundfläche verteilt sind. In der echten Simulationsumgebung sind die Seiten der Unterteilungen gleich lang. Jedoch wurde das Bild für eine bessere Darstellung in der Schriftform gestreckt.

Die Optimierung wurde hier für die Zahl von 5000 Generationen durchlaufen und die Ergebnisse sind in der Paretofront im rechten unteren Teil des Bildes zu erkennen. Der Nutzer der Lighthouse Keeper-Simulation kann hier auf die einzelnen Konfigurationen der Pareto-Optimierung per Maus klicken. Dadurch entsteht im linken Teil die optimalen Beacon-Konfigurationen mit der zu erwartenden Qualität der Lokalisierung. Diese kann als Textdatei mit den Positionsstandorten der Beacon und den einzelnen Vertrauenswerten für jedes Raumelement abgespeichert werden. Zur Veranschaulichung wurde die Werte der berechneten Wahrscheinlichkeiten für gültige Signale farblich in den jeweiligen Raumaufteilungen markiert. Die Farbskala reicht hierbei von blau für einen 100 prozentigen gültigen Empfang, über gelb für 50 Prozent und bis tiefrot für 0 Prozent.

Der Gedanke hinter dieser Aufführung ist es, dass die Signalqualität sichtbar gemacht wird. Unabhängig von der Reichweite bzw. Signalleistung und der Frequenz der Sendeintervalle, will der Anwender eines Lokalisierungssystems die Sicherheit und Genauigkeit der Ortung betrachten können. Mit dieser Simulationsumgebung wird somit nicht nur die Planung realisiert, sondern auch die Machbarkeit eines auf Beacons basierendem Ortungssystems für eine erwünschte Genauigkeit und Dynamik untersucht. Somit kann schon alleine mit Grundrisszeichnungen eine vorläufige Planung durchgeführt werden und sich der Kunde ohne großen Aufwand einen ersten Eindruck über die Leistungsfähigkeit des Systems machen.

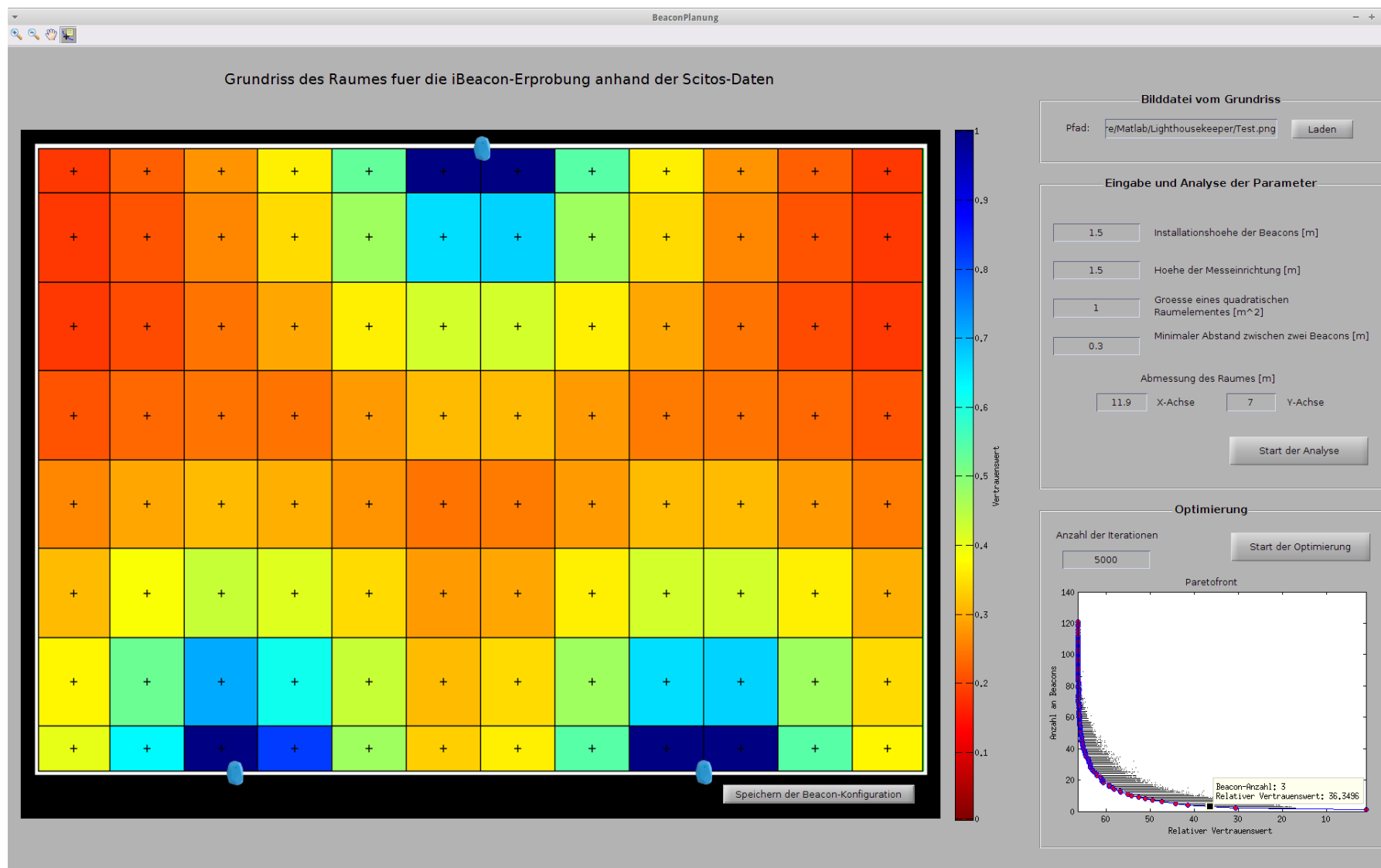


Abb. 1.8: Beispiel einer Lighthouse Keeper-Simulation