Low-Power Positionierung in Sensornetzwerken

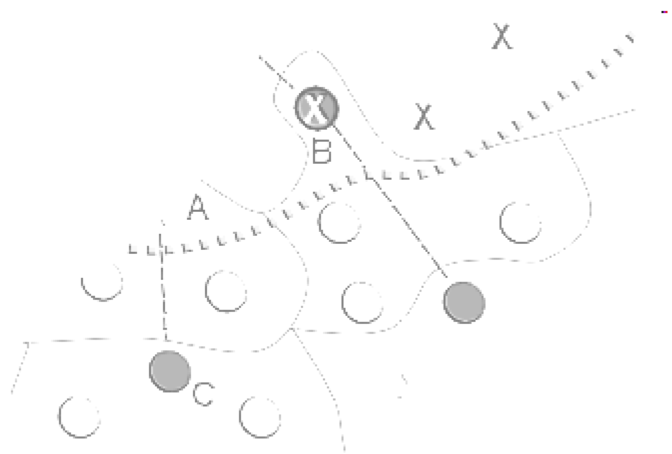
3rd Summerschool, 14.09.2004

Schloss Dagstuhl

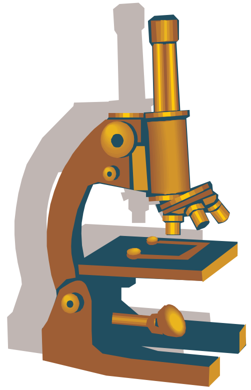
Jan Blumenthal, Universität Rostock

Gliederung

- Einleitung
- Mathematische Methoden
- Komplexitätsbetrachtungen und Energieverbrauch
- Positionsbestimmung in Sensornetzwerken
- Aktuelle Algorithmen



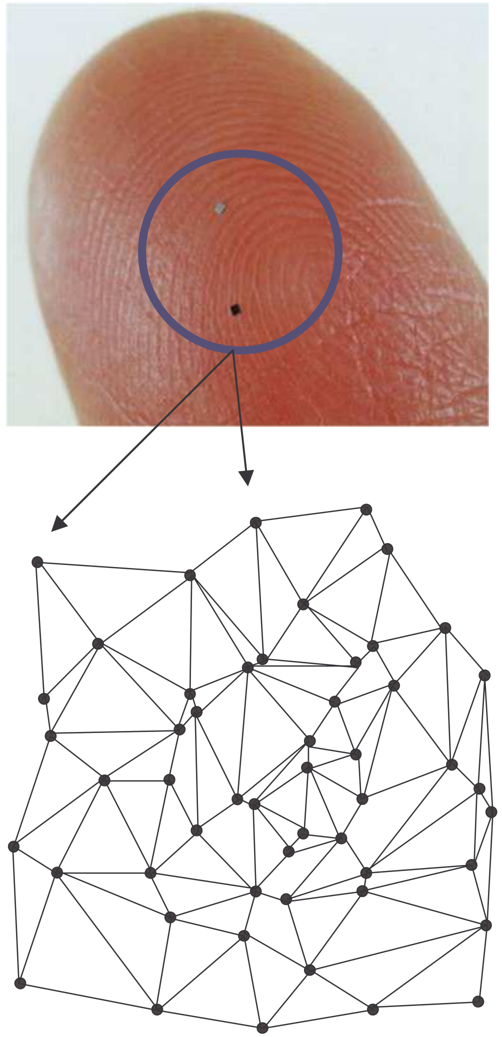
Einleitung



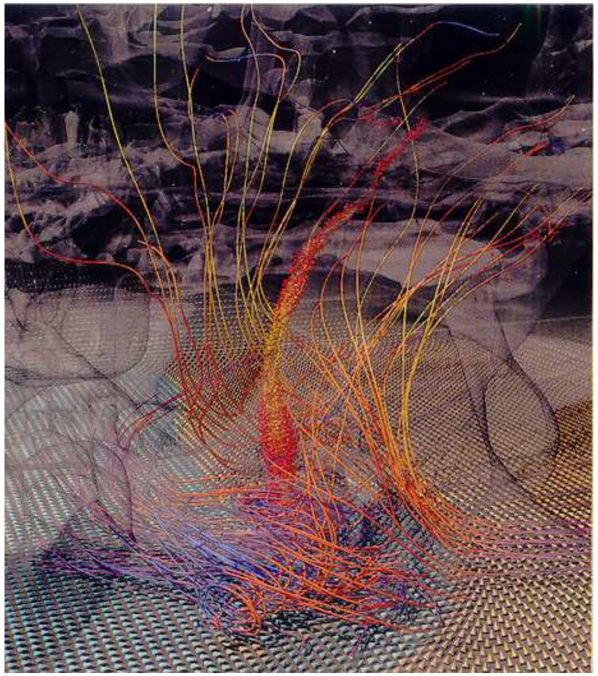
Definition

- Sensornetzwerke:
  - Hunderte Sensorknoten
  - Zufällige Platzierung
  - Drahtlose Kommunikation

- Eigenschaften
  - Sich ändernde Netzwerktopologie
  - Fehleranfälligkeit
  - Selbstorganisierend
  - Ressourcenarm



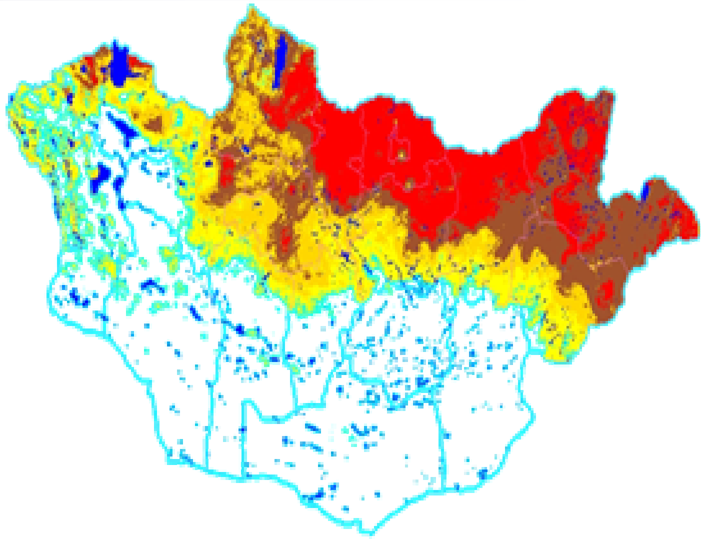
Motivation



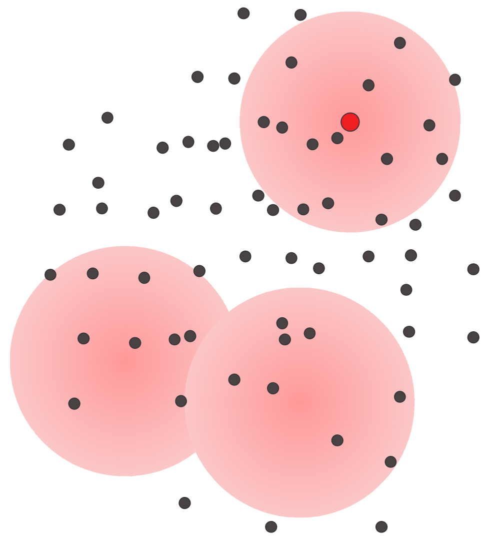
- Analysieren, Beobachten, Entdecken, Überwachen
- z.B. in der Umwelt durch die Überwachung von:
  - Tornados
  - waldbrandgefährdeten Gebieten

Grundlagen

- Warum Positionsbestimmung?
  - Zuordnung: Messung ↔ Ort
  - Effizientes Routing
  - Selbstkonfigurierbarkeit



- Probleme:
  - Zufällige Verteilung der Knoten
  - GPS nicht auf jedem Knoten installierbar



- Lösung:
  - Wenige Knoten mit GPS → Beacons
  - Restliche Knoten → Unbekannte
  - Positionsbestimmung

Mathematische Methoden zur Positionsbestimmung

Einordnung

- Positionierungverfahren
  - Triangulation
    - Lateration
    - Angulation
  - Geometrische Ansätze
    - Schwerpunktbildung
    - Flächenüberlagerung
  - Szenenanalyse
    - Statisch
    - Differentiell
  - Näherungsmessung
    - Physikalischer Kontakt
    - Überwachung von Bezugspunkten

Triangulation

Die Triangulation ist ein Positionsbestimmungsverfahren, bei dem die Punkte nur aus Winkelmessungen zwischen den Punkten ermittelt werden.

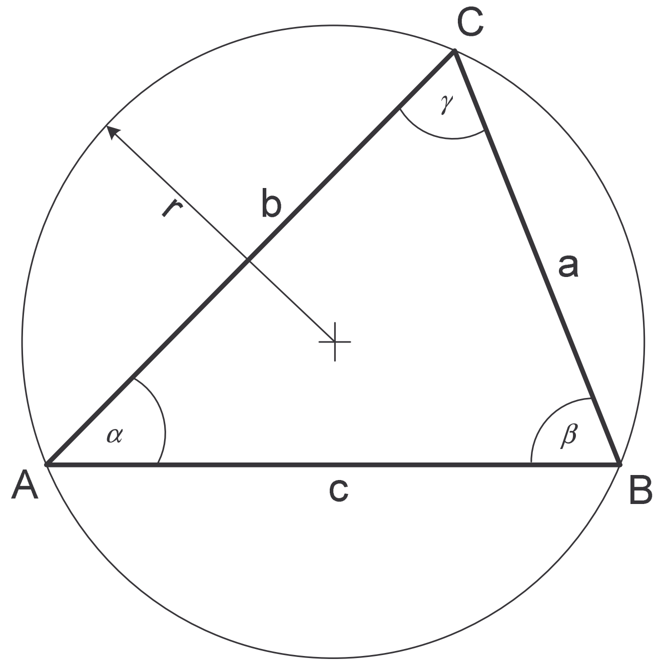
Gleichungen für 3 Knoten im 2D-Fall:

$$\frac{a}{\sin(\alpha)} = \frac{b}{\sin(\beta)} = \frac{c}{\sin(\gamma)}$$

$$a^2 = b^2 + c^2 - 2 \cdot b \cdot c \cdot \cos(\alpha)$$

$$b^2 = a^2 + c^2 - 2 \cdot a \cdot c \cdot \cos(\beta)$$

$$c^2 = a^2 + b^2 - 2 \cdot a \cdot b \cdot \cos(\gamma)$$



Fazit:

- 3 Messwerte (α, β, γ) erforderlich
- 1 Position (A, B oder C) sinnvoll
- Maßstab
- Gleichungssystem eindeutig bestimmbar

Trilateration

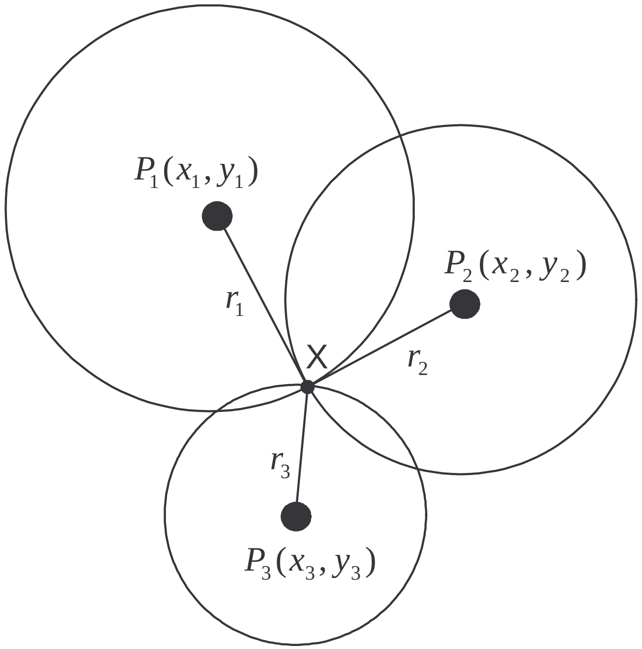
Die Trilateration ist ein Positionsbestimmungsverfahren, bei dem die Punkte nur durch Streckenmessungen bestimmt werden.

Gleichungen für 3 Knoten im 2D-Fall:

$$\sqrt{(x_1 - x)^2 + (y_1 - y)^2} = |\overrightarrow{r_1}|$$

$$\sqrt{(x_2 - x)^2 + (y_2 - y)^2} = |\overrightarrow{r_2}|$$

$$\sqrt{(x_3 - x)^2 + (y_3 - y)^2} = |\overrightarrow{r_3}|$$



Fazit:

- 3 Messwerte (r_1...r_3) erforderlich
- 3 Positionen (P_1...P_3) gegeben
- 1 absolute Position sinnvoll
- Gleichungssystem ist eindeutig bestimmbar

Besonderheiten in Sensornetzwerken

Sensornetzwerk mit vielen Knoten

- Alle Knoten erhalten Informationen über Nachbarn, z.B. durch Distanzmessungen
- Jeder Knoten benötigt lediglich eigene Position (2 Werte)

- Anzahl Eingangsdaten>>Anzahl Ausgangsdaten
- Überbestimmtes Gleichungssystem

↓

Gleichungssystem explizit nicht lösbar!

↓

Gleichungssystem nur näherungsweise lösbar!

↓

Ausgleichsrechnung

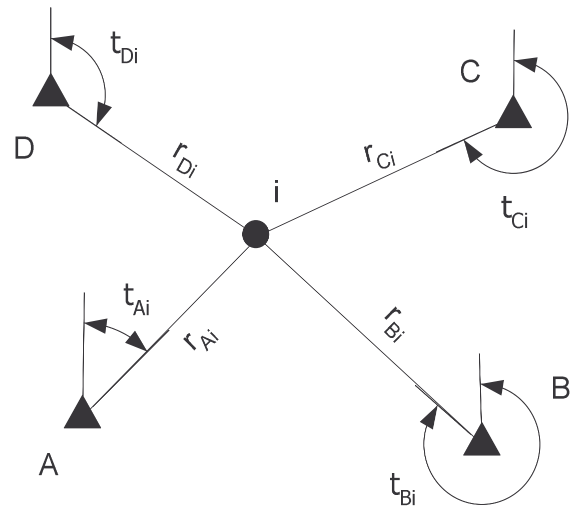
Überbestimmte Trilateration (1 aus 4)

Sensornetzwerk mit 5 Knoten

Geg: • 4 bekannte Nachbarn $P_k(x_k,y_k)$ mit k=A,B,C,D

• Strecken r_{ij} zwischen P_k und P_i

Ges: • Unbekannte Position $P_i(x_i,y_i)$



Basisbeziehungen

$$r_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

$$b_{ij} = r_{ij_{measured}} - r_{ij_{\mathrm{determined}}}$$

Lösungsansatz

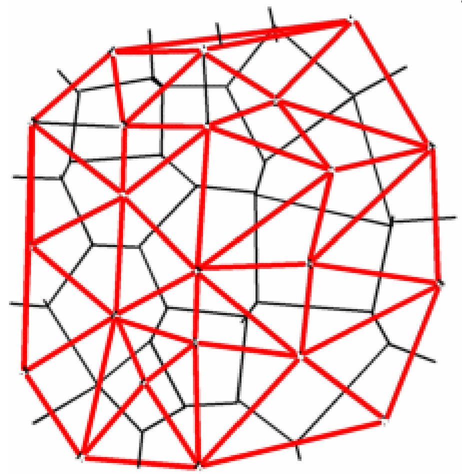
- Matrix aus Gleichungen aufstellen

$$A \cdot x = b \quad \Longrightarrow \quad x = (A^T A)^{-1} A^T b$$

Überbestimmte Trilateration (n aus m)

Sensornetzwerk mit m+n Knoten:

- Gleichzeitige Berechnung von mehreren Positionen
- Gegenseitige Abhängigkeit zwischen unbekannten Positionen
- Redundanzausnutzung



Ansatz:

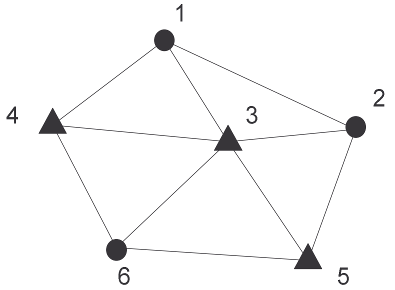
- Lösungsformel erweitern

$$x = \left(A^T A\right)^{-1} A^T b$$

Problem:

- Zentrale Berechnung
- Berechnung wird noch komplexer
- Zeit- und kommunikationsaufwändig

Überbestimmte Trilateration (n aus m)



● : Knoten mit bekannter Position

▲ : Knoten mit unbekannter Position

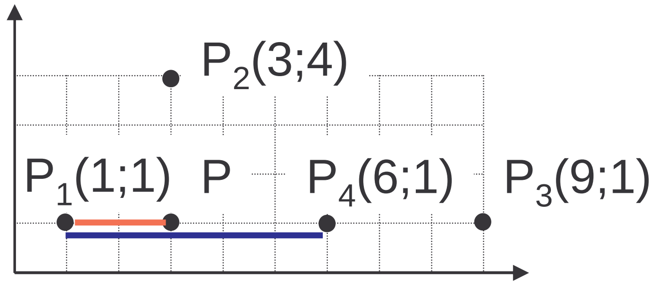
Resultierende Matrix:

- x=Verbindung (symbolisiert math. Term)
- 0=keine Verbindung
- Riesige Matrix
- Matrix symmetrisch dünn besetzt!

	X_1	Y_1	X_2	Y_2	X_3	Y_3	X_4	Y_4	X_5	Y_5	X_6	Y_6
X_1	x	x	x	x	x	x	x	x	0	0	0	0
Y_1		x	x	x	x	x	x	x	0	0	0	0
X_2			x	x	x	x	0	0	x	x	0	0
y_2				x	x	x	0	0	x	x	0	0
X_3					x	x	x	x	x	x	x	X
Y_3						x	x	x	x	x	x	X
X_4							x	X	0	0	x	x
y_4								x	0	0	x	x
X_5									x	x	x	X
Y_5										x	x	X
X_6											X	X
Y_6												X

Beispiel:

Geg.:



Ges: Position von P

- 4 Messwerte ($r_1 \ldots r_4$)
- 4 Knotenpositionen ($x_1 \ldots x_4, y_1 \ldots y_3$)
- 2 Unbekannte (x,y)
- r_i=Abstand von P_i zu P
- d_{i1}=Abstand von P_i zu P_1

P_i	x_i	y_i	r_i	d_{i1}
1	1	1	2	0
2	3	4	3	$\sqrt{13}$
3	9	1	6	8
4	6	1	4	5

Herleitung des Gleichungssystems

$$\sqrt{(x-x_i)^2+(y-y_i)^2} = |\vec{r_i}|$$

$$(x-x_i)^2+(y-y_i)^2 = r_i^2$$

$$(x-x_j+x_j-x_i)^2+\ldots = r_i^2$$

$$((x-x_j)-(x_i-x_j))^2+\ldots = r_i^2$$

$$(x-x_j)^2-2(x-x_j)(x_i-x_j)+(x_j-x_i)^2+\ldots = r_i^2$$

$$(x-x_j)(x_i-x_j)+\ldots = -\frac{r_i^2-(x-x_j)^2-(x_j-x_i)^2+\ldots}{2}$$

Abstandsgleichung

$P_j(x_j,y_j)$=bekannter Referenzpunkt

$+x_j - x_j$

Klammern setzen: $a=x-x_j$, $b=x_j-x_i$

Bin. Formeln: $(a-b)^2=a^2-2ab+b^2$

$$r_j^2 = (x-x_j)^2+(y-y_j)^2$$

$$d_{ij}^2 = (x_j-x_i)^2+(y_j-y_i)^2$$

$$\Rightarrow \quad \ldots = \frac{r_j^2-r_i^2+d_{ij}^2}{2}$$

Herleitung des Gleichungssystems II

Allgemeine Form:	$(x-x_j)(x_i-x_j)+(y-y_j)(y_i-y_j)=\dfrac{r_j^2-r_i^2+d_{ij}^2}{2}$
Referenzpunkt j=1:	$(x-x_1)(x_i-x_1)+(y-y_1)(y_i-y_1)=\dfrac{r_1^2-r_i^2+d_{i1}^2}{2}$
Gleichungssystem für Beispiel:	$(x-x_1)(x_2-x_1)+(y-y_1)(y_2-y_1)=\dfrac{r_1^2-r_2^2+d_{21}^2}{2}$ $(x-x_1)(x_3-x_1)+(y-y_1)(y_3-y_1)=\dfrac{r_1^2-r_3^2+d_{31}^2}{2}$ $(x-x_1)(x_4-x_1)+(y-y_1)(y_4-y_1)=\dfrac{r_1^2-r_4^2+d_{41}^2}{2}$

Herleitung des Gleichungssystems III

Allg. Gleichung für für Beispiel:	$(x-x_1)(x_i-x_1)+(y-y_1)(y_i-y_1)=\frac{r_1^2-r_i^2+d_{i1}^2}{2}$
Matrixschreibweise:	$\underbrace{\begin{vmatrix} x_2-x_1 & y_2-y_1 \\ x_3-x_1 & y_3-y_1 \\ x_4-x_1 & y_4-y_1 \end{vmatrix}}_{\underline{\mathbf{A}}} \bullet \underbrace{\begin{vmatrix} x-x_1 \\ y-y_1 \end{vmatrix}}_{\underline{\mathbf{x}}} = \underbrace{\frac{1}{2}\begin{vmatrix} r_1^2-r_2^2+d_{21}^2 \\ r_1^2-r_3^2+d_{31}^2 \\ r_1^2-r_4^2+d_{41}^2 \end{vmatrix}}_{\underline{\mathbf{b}}}$
Zu lösendes Gleichungssystem:	$\begin{vmatrix} 2 & 3 \\ 8 & 0 \\ 5 & 0 \end{vmatrix} \bullet \begin{vmatrix} x-x_1 \\ y-y_1 \end{vmatrix} = \begin{vmatrix} 4 \\ 16 \\ 10 \end{vmatrix}$

Methode der kleinsten Quadrate (MdkQ)

$$Ax = b$$
$$Ax - b = 0$$
$$r = Ax - b$$

Allg. Lösung für quadratische Matrizen mit expliziter Lösung

r=Residuenvektor für überbestimmte Gleichungssysteme

r=Ax-b muß orthogonal auf Spaltenraum von A liegen bzw. äquivalent in Nullraum A^T.

Skalarprodukt aus A^T und r gleich Null setzen, dann steht A^T senkrecht auf r.

$$A^T r = 0$$
$$A^T (Ax - b) = 0$$
$$A^T Ax - A^T b = 0$$
$$A^T Ax = A^T b$$

$$x = \left(A^T A\right)^{-1} A^T b$$

Allg. Lösungsformel für überbestimmte Gleichungssysteme

Methode der kleinsten Quadrate II

Ansatz: $x = \left(A^T A\right)^{-1} A^T b$

Rechenschritt	Bemerkung	Aufwand
Rang bestimmen	Gleichungssystem lösbar ?	Gleichungssystem (Gauss)
Berechnung von A^TA	-	Matrix-Multiplikation
Inverse bilden $(A^TA)^{-1}$	Mehrdimensionale Determinanten	n Gleichungssysteme
$(A^TA)^{-1}A^T$	-	Matrix-Multiplikation
$(A^TA)^{-1}A^Tb$	-	Matrix-Multiplikation
Koordinaten extrahieren	-	n Gleichungen
Fehlerabstand	Residuenvektor	Matrix-Multiplikation

Methode der kleinsten Quadrate III

Bemerkungen:

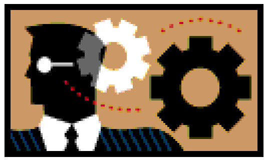
+ Berechnung nach Schema
+ Genauigkeit steigt mit Anzahl der Eingangswerte
+ Rel. geringer Speicherverbrauch ($\dim(A^T A)=n$)
- Berechnung sehr komplex
- Benötigt FPU
- Numerisch instabil (auch bei Floats)
  - Matlababweichung bei exakten Eingangswerten: 2,0 vs. 2,004
- Komplizierte Berechnung der Inversen bei großen Matrizen

Alternativen zur MdkQ

Alternativen bzw. konvergenzverbessernde Verfahren

- Pivotisierung
- LR(LU)-Faktorisierung
- Singulärwertmethode
- Cholesky-Zerlegung
- QR-Zerlegung
- Gram-Schmidt-Verfahren
- Householder-Spiegelung
- Iterative Verfahren
  - Kalman Filter

Komplexitätsbetrachtungen und Energieverbrauch

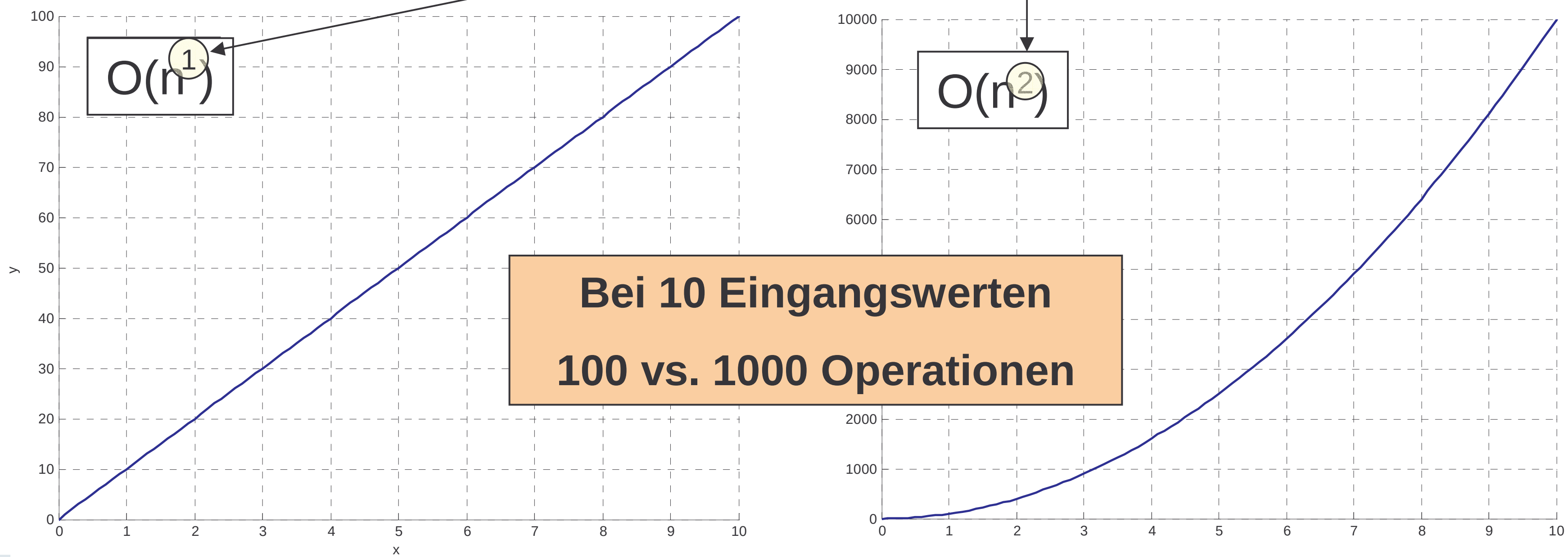


Komplexität numerischer Algorithmen

- Geschwindigkeitsangabe von Algorithmen
- Gibt die Anzahl der Operationen eines Algorithmus in Abhängigkeit der Eingangsparameter *n* an
- Formelzeichen: *O* (dimensionslos)

Beispiel

Ordnung des Algorithmus



Berechnung der Komplexität

- Zählen aller Gleitpunktoperationen
- Wertigkeit einer Operation: 1

Addition zweier Vektoren

```
z=zeros(n,1);
for i=1:n
    z(i) = x(i) + y(i);
end
```

n Additionen

O(n)

Skalarprodukt zweier Vektoren

```
z=0;
for i=1:n
    z(i) = z + x(i) + y(i);
end
```

n Additionen

n Multiplikationen

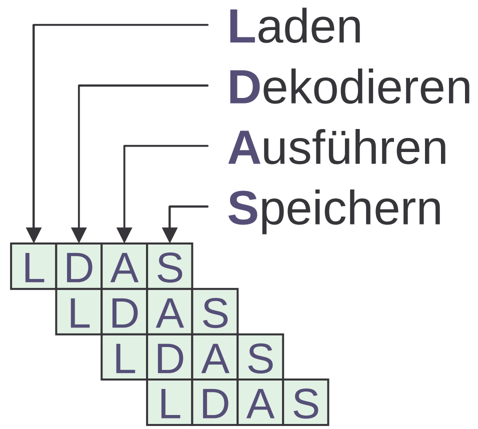
O=(2*n)=n

Ordnungskonstante

Komplexität einzelner Verfahren

Anwendung	Operationen	Komplexität
Vektoraddition	n Additionen	O(n)
Skalarprodukt	2*n	O(n)
Lösen eines linearen Gleichungssystems (Gauss)	$\left(\frac{1}{2}n^2 - \frac{1}{2}n\right) + \left(\frac{1}{2}n^2 - \frac{1}{2}n\right) + (n)$	$O(n^2)$
Matrix Multiplikation	$n^{3+}n^2$	$O(n^3)$
LU-Faktorisierung	$\frac{2}{3}n^3$	$O(n^3)$

Komplexität und Leistungsverbrauch



Skalare Prozessoren

- 1 Operation pro Takt durch Pipelining
- Anzahl der Operationen bestimmt zu benötigenden Takte

Energieverbrauch

$$P = C_L \cdot V_{DD}^2 \cdot f + P_{Leak}$$

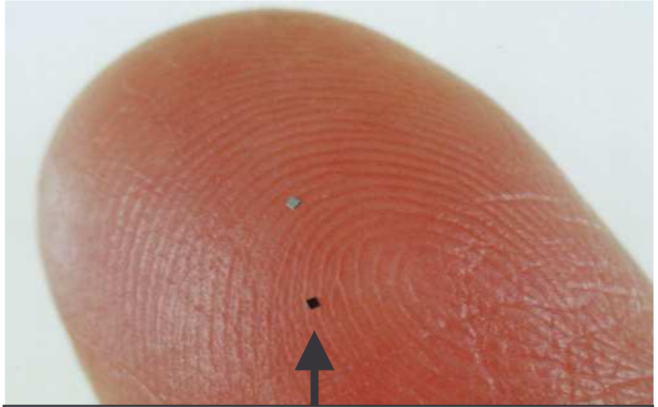
C_L = Lastkapazität
V_{DD} = Betriebsspannung
f = Taktfrequenz
n = Anzahl der Operationen

Günstigster Energieverbrauch (ohne Stalls)

$$P = C_L \cdot V_{DD}^2 \cdot n$$

Implikationen der Kommunikation

- **Energieverfügbarkeit :** ~1 J/mm³ **einer Batterie**
- **Moorsch'es Gesetz für Batterien nicht absehbar**
- **Max. 2-20% Kapazitätserhöhung der Batterien pro Jahr**

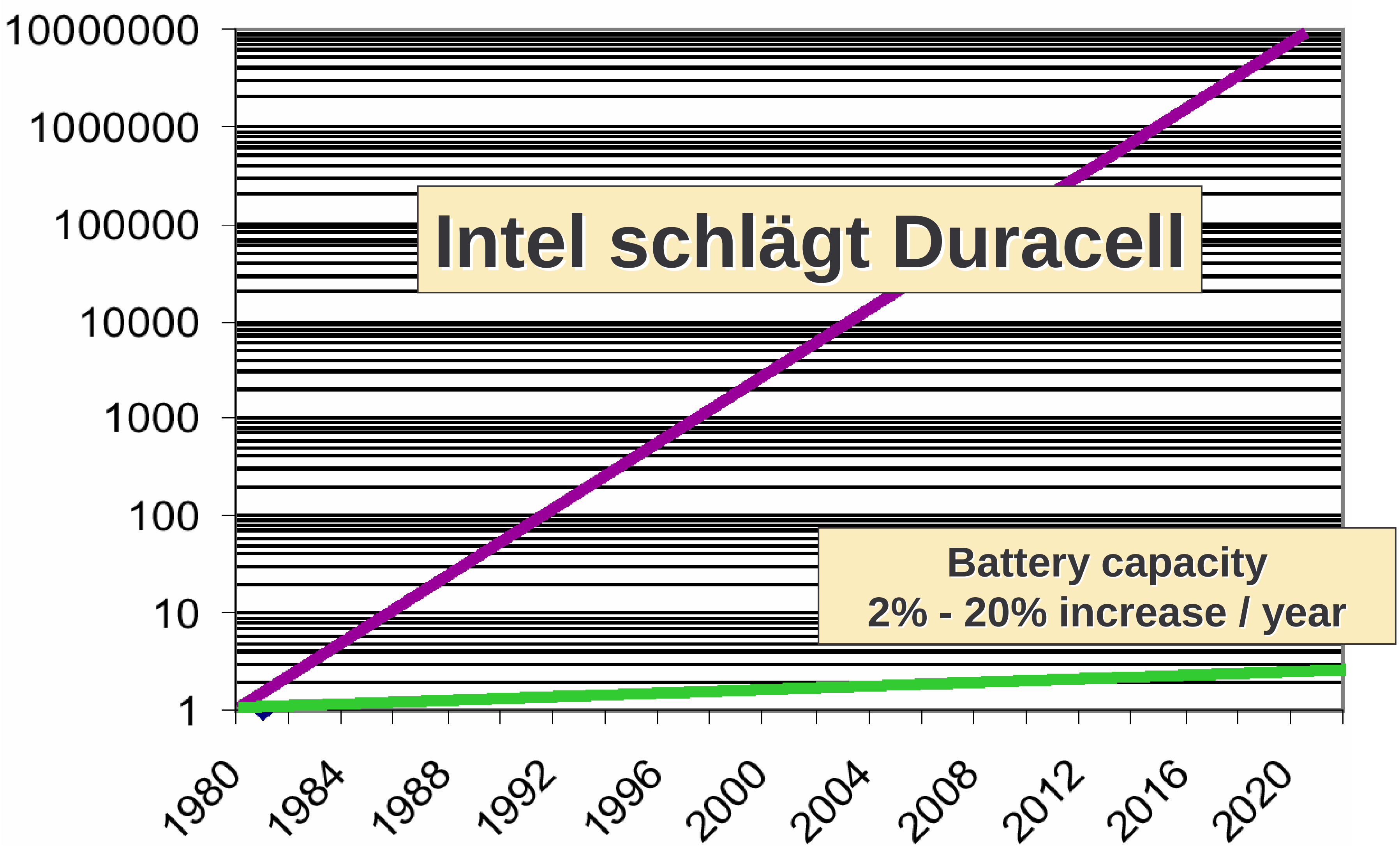


Zukünftige aktive Sensorknoten

Rechnen ist ressourcenschonender als kommunizieren

Pentium 4	26,8 nJ/Operation	2,8 GHz@74,9 W
ARM7TDMI	0,06 nJ/Operation	133 MHz@8 mW
Chipcon CC1010	3,31 nJ/Operation	14,74 MHz@48mW
Atmel ATmega 128	4,1250 nJ/Operation	4 MHz@16.5 mW
Bluetooth (10 m)	90 nJ / Bit	brutto 1 MBit/s@90 mW
Bluetooth (100 m)	500 nJ / Bit	brutto 1 MBit/s@500 mW

Trends



10000000
1000000
100000
10000
1000
100
10
1
1980
1984
1988
1992
1996
2000
2004
2008
2012
2016
2020
Intel schlägt Duracell
Battery capacity
2% - 20% increase / year

Komplexität bei Mobilität

Geschwindigkeitsbestimmung:

- 2 Positionen
- 1 Zeitmessung

$$v_i = \frac{x_2 - x_1}{t}$$

$$\boxed{O_v = \underbrace{(O(n^3) + O(n^3))}_{x_2 - x_1} \cdot \underbrace{O(n)}_{t} = n^4}$$

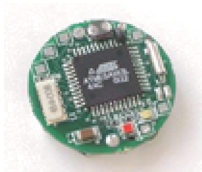
Komplexität steigt um mindestens eine Potenz!

Berechnung:

- Methode der kleinsten Quadrate
- Kalman-Filter

Komplexität in Sensornetzwerken

- ChipCon CC1010:
  - 8051 Mikrocontroller
  - Takt: Bis zu 16MHz → 16 MFlops
- Atmel ATmega 128 (Mica Mote)
  - Takt: 4 MHz → 4 MFlops



Beispiel: Prozessor mit 10 MFlops

n	O(n)	O(n^2)	O(n^3)	O(n^4)
10	10^{-6} s	10^{-5} s	10^{-4} s	10^{-3} s
100	10^{-5} s	10^{-3} s	10^{-1} s	10 s
1000	10^{-4} s	10^{-1} s	10^{2} s (≈ 1.7 min)	10^{5} s (≈ 1.2 Tage)
10000	10^{-3} s	10^{1} s	10^{5} s (≈ 1.2 Tage)	10^{9} s (≈ 31 Jahre)

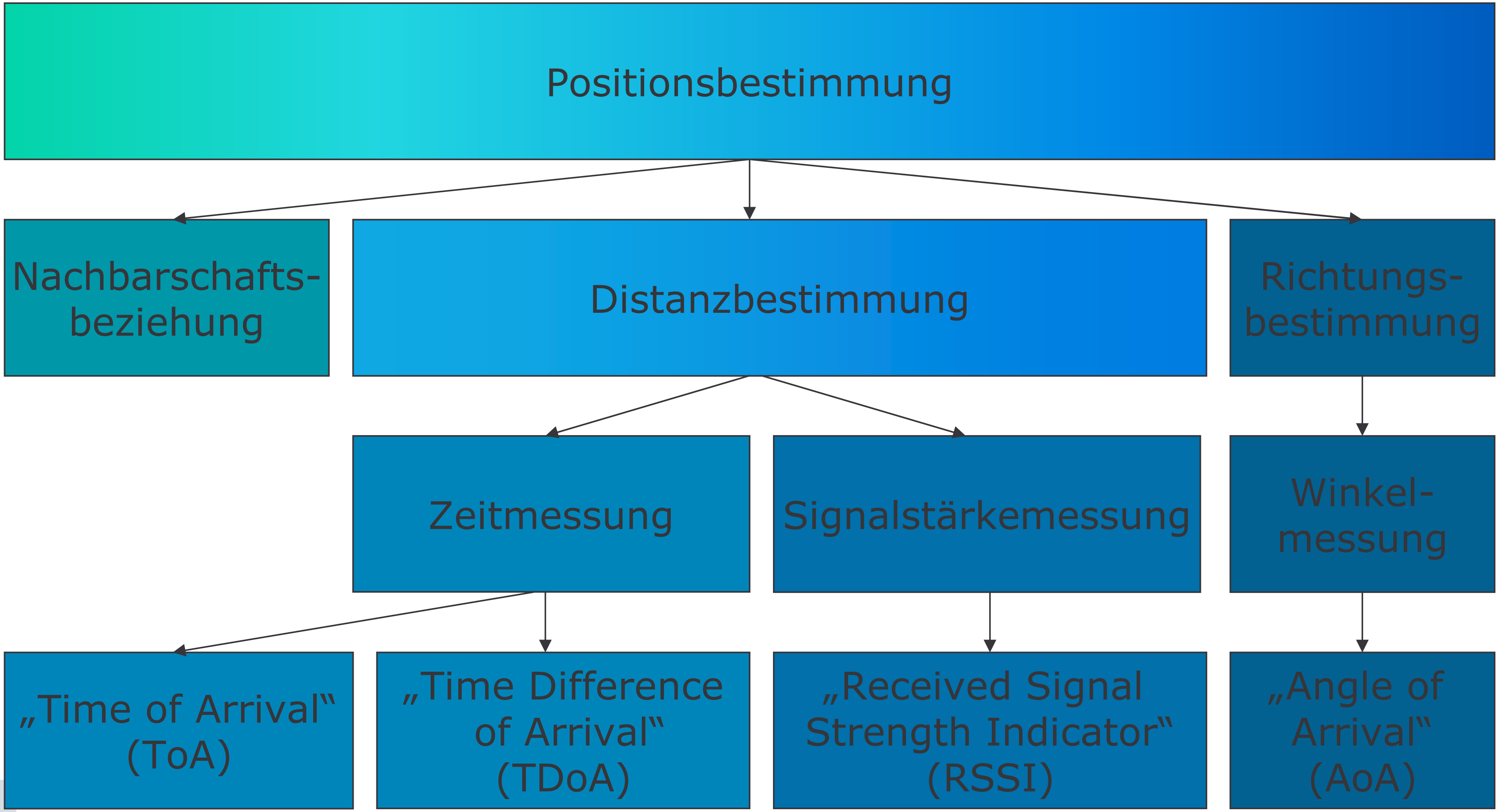
Aber: Diese Prozessoren haben keine Floating Point Unit (FPU) !

Zusammenfassung

- Exakte mathematische Verfahren kaum nutzbar, weil
  - Keine FPU vorhanden
  - Komplexität der Algorithmen zu hoch
  - Hohe Kommunikation
  - Riesiger Speicherverbrauch durch Matrizen
  - Hoher Energieverbrauch

Positionierung in Sensornetzwerken

Klassifikation der Positionsbestimmung



Time of Arrival

- Synchronisation zwischen Sender und Empfänger erforderlich
- Zeitmessung erforderlich

Beispielrechnung

- Abstand zwischen 2 Sensorknoten ***d=30cm***

$$v_0 = 300000 \frac{km}{s}$$

$$v_{Luft} \approx \frac{2}{3} v_0 \approx 200000 \frac{km}{s}$$

$$\frac{200000 km}{1 s} = \frac{30 cm}{t_d}$$

$$t_d = 1,5^{-9} s = 1,5 ns$$

$$\boxed{f_d = 2 \frac{1}{1,5^{-9} s} = 1.333 GHz}$$

Fazit: Time of Arrival per Funk in Sensornetzwerken nicht anwendbar!

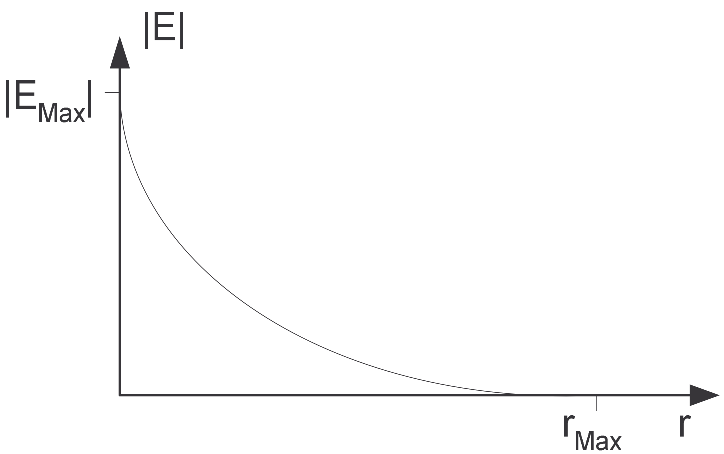
Positionsbestimmung mit RSSI (ideal)

RSSI = Received Signal Strength Indicator

Berechnung:

$$r = \begin{cases} \sqrt{\dfrac{r_{Max}^{\;2}\left(\left|E_{Max}\right| - \left|E_{Measured}\right|\right)}{\left|E_{Max}\right|}} & 0 < \left|E_{Measured}\right| < \left|E_{Max}\right| \\ \textit{nicht definiert} & \textit{sonst} \end{cases}$$

Idealer Signalfeldstärkeverlauf (Empfang):

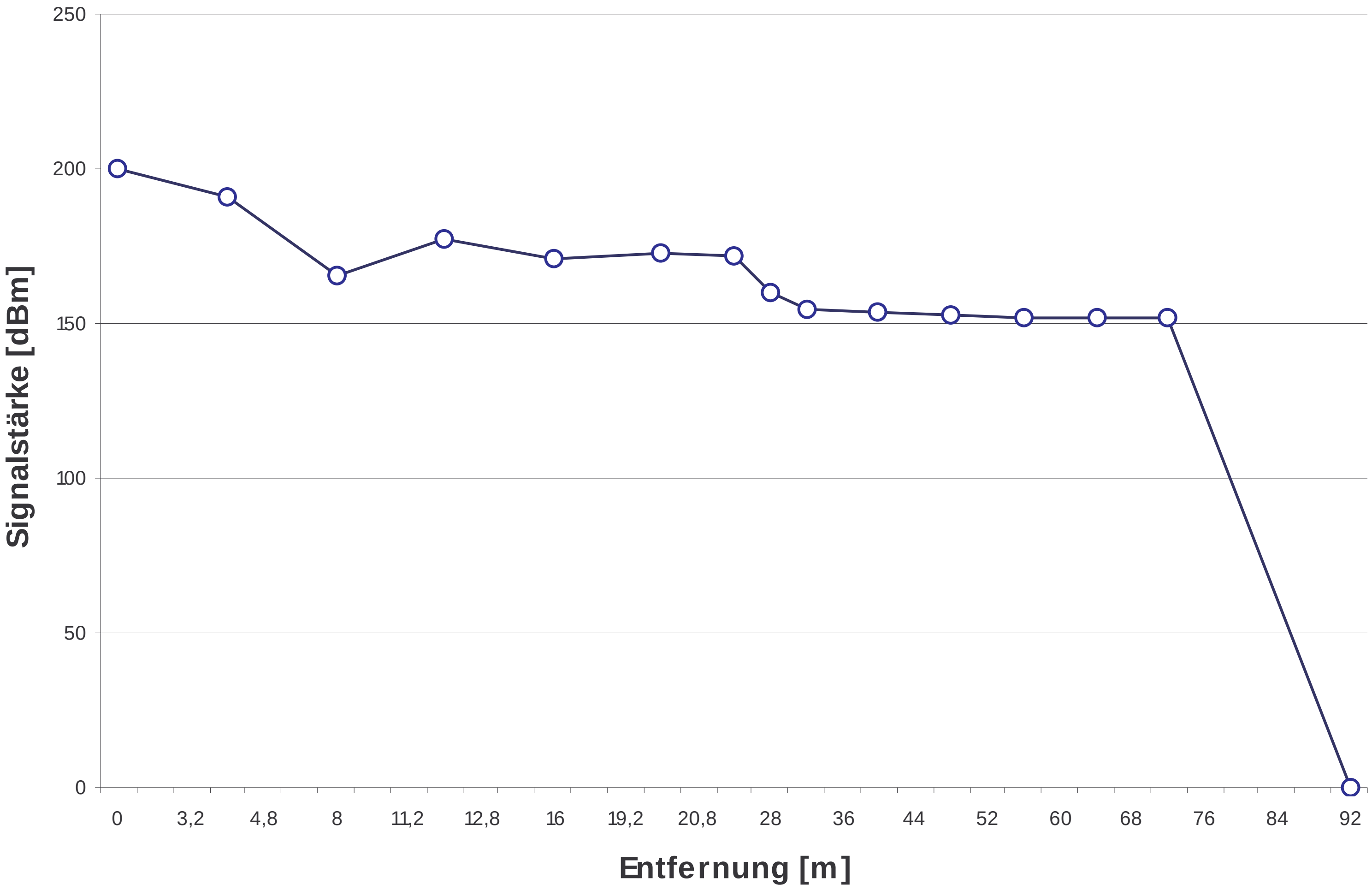


$\left|\vec{E}\right|$ = Signalfeldstärke (RSSI)

r = Entfernung zum Sender

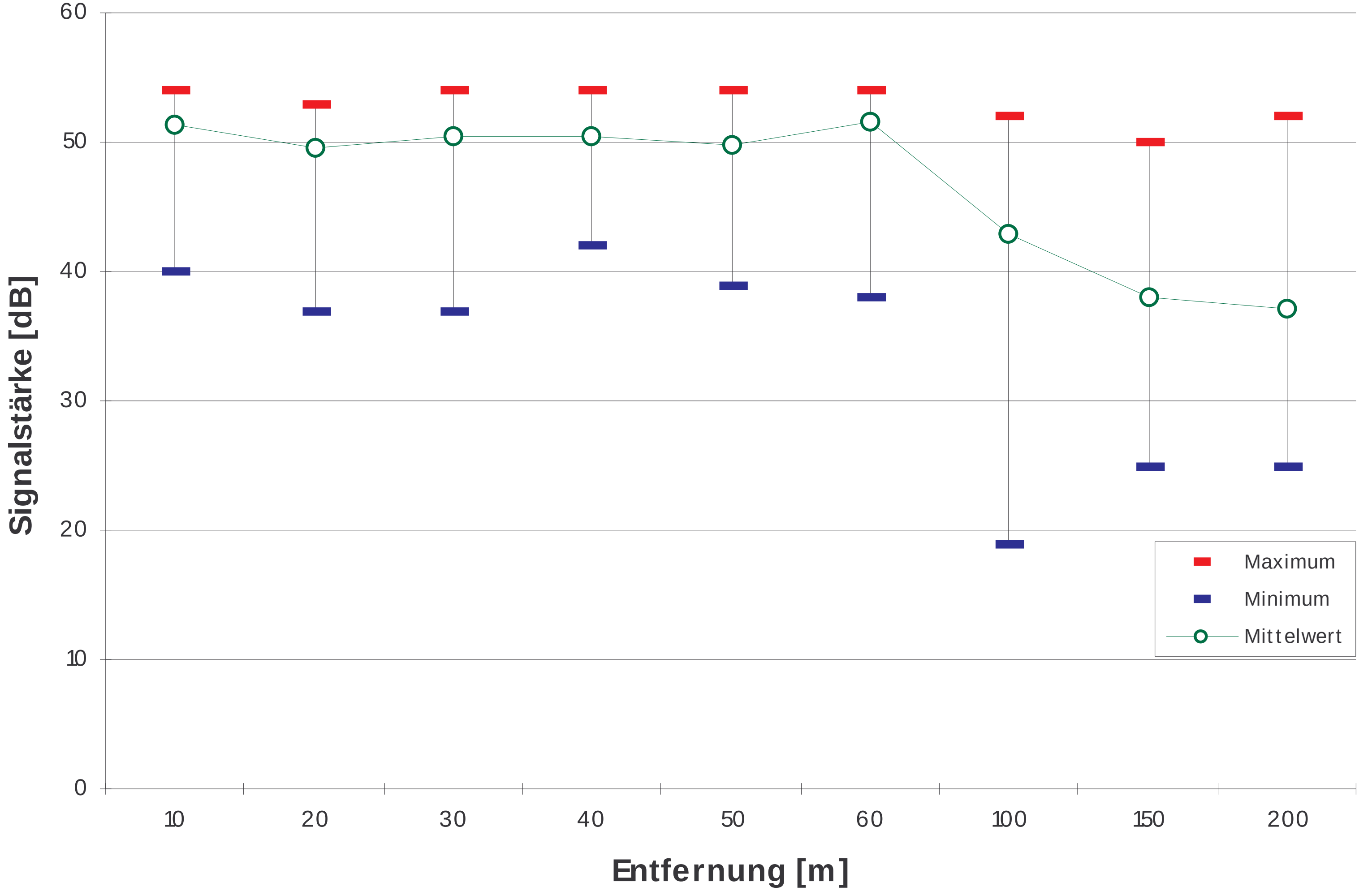
r_{Max} = Max. Übertragungsreichweite

Signalstärkemessungen Chipcon



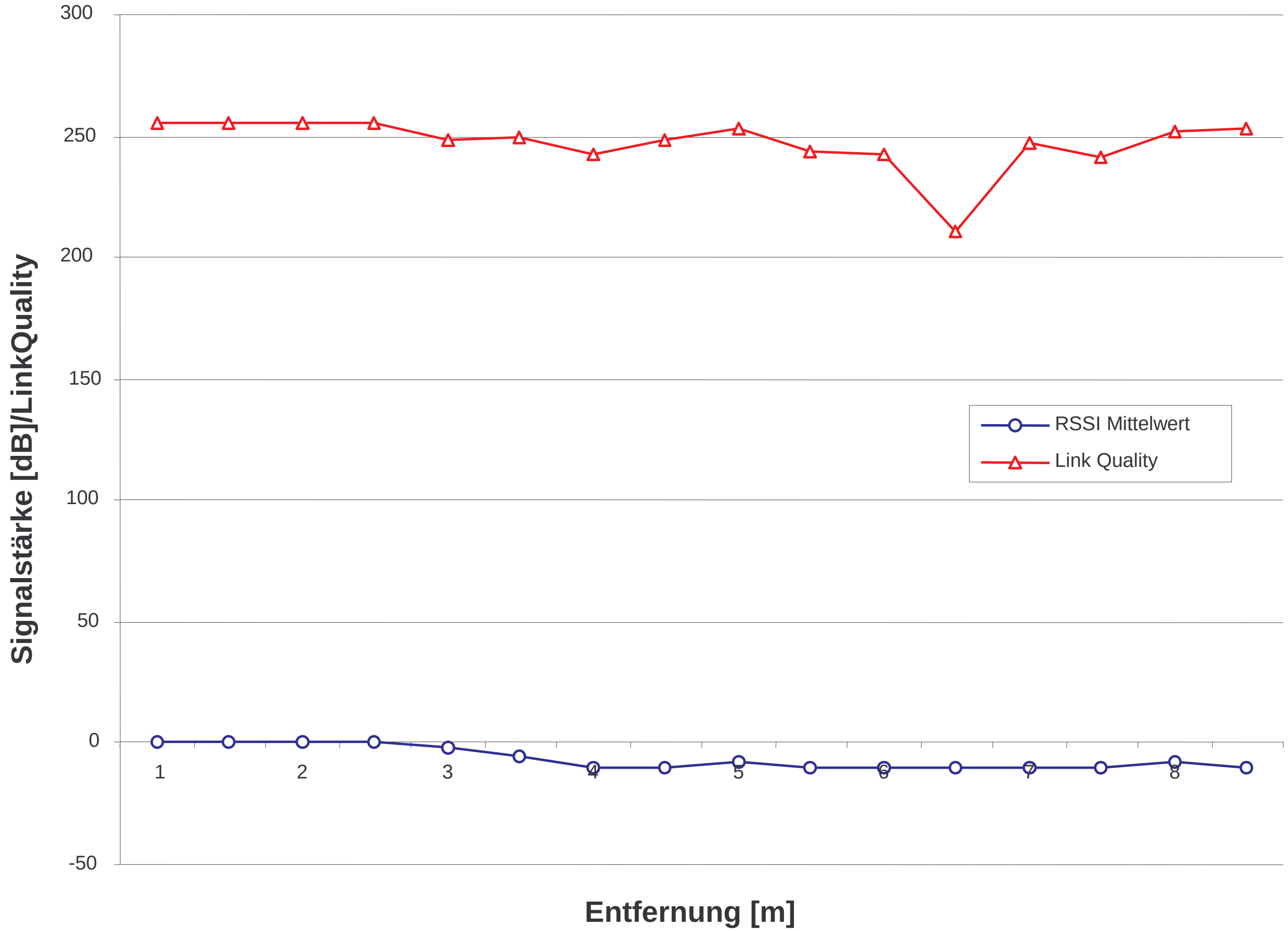
➡ Received Signal Strength Indicator unbrauchbar !

Signalfeldstärkemessungen WLAN



Received Signal Strength Indicator unbrauchbar !

Signalfeldstärkemessungen Bluetooth



Received Signal Strength Indicator unbrauchbar !

Bewertungskriterien

Eigenschaft	Bewertungskriterien	
	Wunsch	Realität
Genauigkeit der Eingangs-werte (z.B. RSSI)		
Genauigkeit der ermittelten Positionswerte		[1]
Energieverbrauch		[1]
Verhalten in Extremsituationen		[1]
Skalierung für große Netzwerke		[1]
Anzahl Knoten mit unbestimmbarer Position		[1,2]

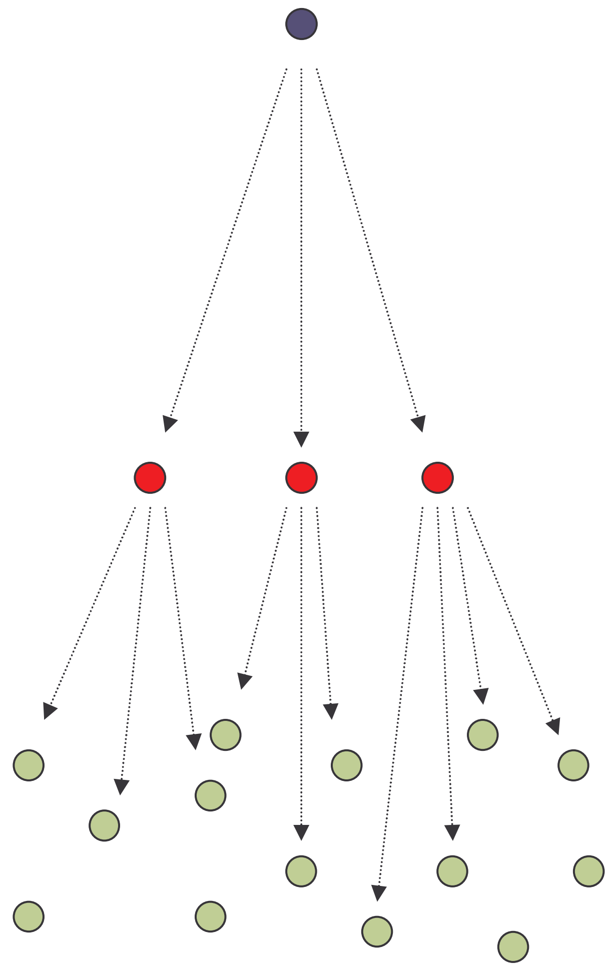
1) Abhängig vom verwendeten Positionierungsverfahren
2) Abhängig von eingestellten Parametern

Hierarchie in Sensornetzwerken

Basisstation/Gateway:
- Feste Station
- Eigene Position bekannt
- Ressourcen unkritisch

Beacon:
- Drahtlose Knoten im Netzwerk
- **Eigene Position bekannt**
- Knappe Ressourcen

Einfache Sensorknoten:
- Drahtlose Knoten im Netzwerk
- **Eigene Position nicht bekannt**
- Sehr knappe Ressourcen



: Basisstation   : Beacon   : Einfacher Sensorknoten

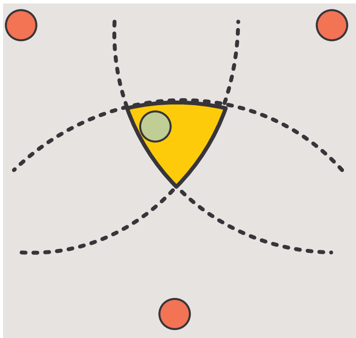
Aktuelle Algorithmen

Positionierungsalgorithmen

Positionsbestimmung

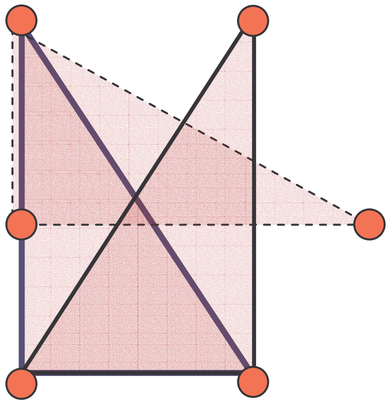
Approximierte: Δ>5%

Exakte: Δ<5%

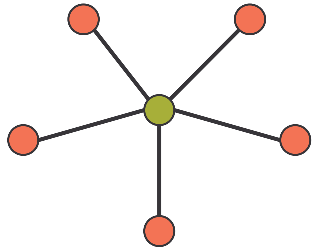


$$X_{Est} = \left(\frac{X_{i1} + \ldots X_{ik}}{k} \right)$$

Mittelpunktbildung
(Coarse Grained)



Dreiecksbildung
(APIT)



$$\Delta_i(x_0, y_0, d_i) = d_i - \sqrt{(x_i - x_0)^2 - (y_i - y_0)^2}$$

$$\underline{b} = (\underline{X}^T \underline{X})^{-1} \underline{X}^T \underline{y}$$

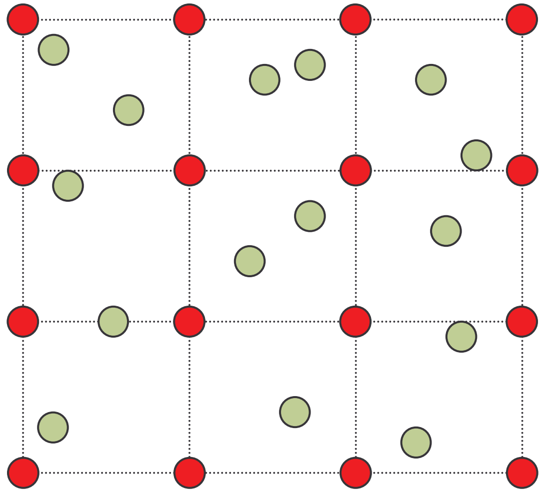
Überbest. Trilateration
(Atomar, iterativ, kollaborativ)

Δ : Fehler in der Berechnung der Position

● : Beacon

● : Einfacher Knoten

Coarse Grained: Modell



Voraussetzungen:

- Quadratische Anordnung der Beacons
- Beacon in Gitternetzanordnung (Infrastruktur-Verteilung)
- Kreisförmige Übertragungsreichweite
- Zufällig verteilte Sensorknoten

Ziele:

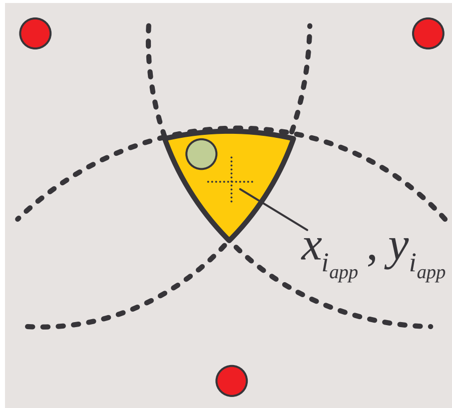
- Kleiner Positionierungsfehler
- Geringer Energieverbrauch
- Geringe Anzahl von Beacons
- Einfache Simulation und analytische Betrachtungen

: Beacon : Sensor Node

Funktionsweise: Coarse Grained (CGL)[1]

Positionsbestimmung:

- Knoten empfängt Nachrichten mit Positionen von n verschiedenen Beacons
- Positionsberechnung durch einfache Mittelpunktbildung
- Sensorknoten kann nur in Bereich liegen, der von allen Beacons erreicht wird



Vorteile:

- Keine RSSI Werte
- Keine Kommunikation durch einfache Knoten (nur Empfang)
- Einfache Berechnung

$$x_{i_{app}}, y_{i_{app}} = \left(\frac{1}{n} \sum_{k=1}^{n} x_{B_k}, \frac{1}{n} \sum_{k=1}^{n} y_{B_k} \right)$$

: Beacon : Einfacher Knoten

: Zielgebiet, in dem Knoten liegt

: Übertragungsreichweite der Beacons

[1]) N. Bulusu, et. al.: *„GPS-less Low Cost Outdoor Localization For Very Small Devices“*

Positionierungsfehler in CGL

Positionierungsfehler:

- Distanz zwischen bestimmter Position und wahrer Position

$$f_i(x,y) = \sqrt{(x_{i_{app}} - x_{i_a})^2 + (y_{i_{app}} - y_i)^2}$$

$x_{i_{app}}, y_{i_{app}}$ = Bestimmte Koordinaten von Knoten i

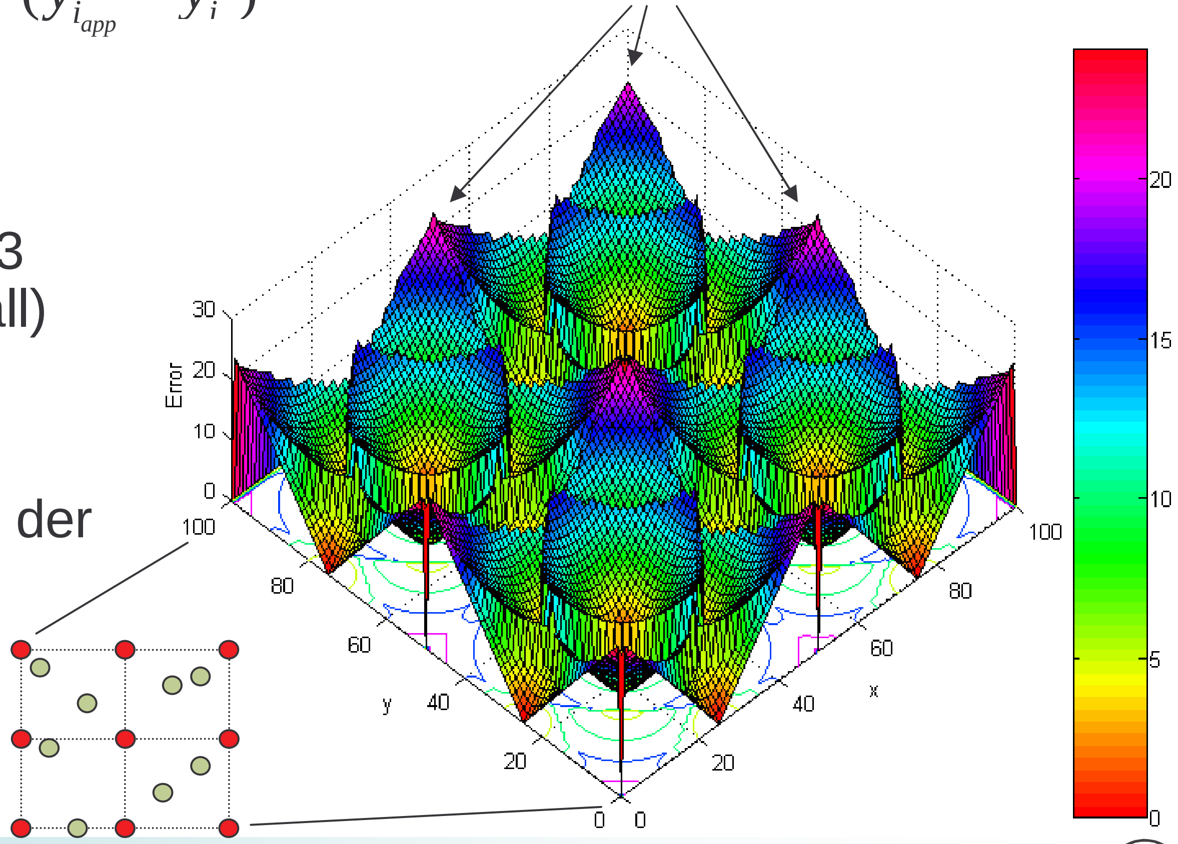
x_{i_a}, y_{i_a} = Exakte Koordinaten von Knoten i

f_i = Positionierungsfehler von Knoten i

● = Beacons

Fehlerverlauf:

- Gitteranordnung von 3x3 Beacons (Infrastrukturfall)
- Feldbreite 100x100
- 101x101 Sensorknoten
- Übertragungsreichweite der Beacons r=50



Grenzwertbetrachtungen

Positionierungsfehler maximal, wenn Übertragungsreichweite *r* der Beacons:

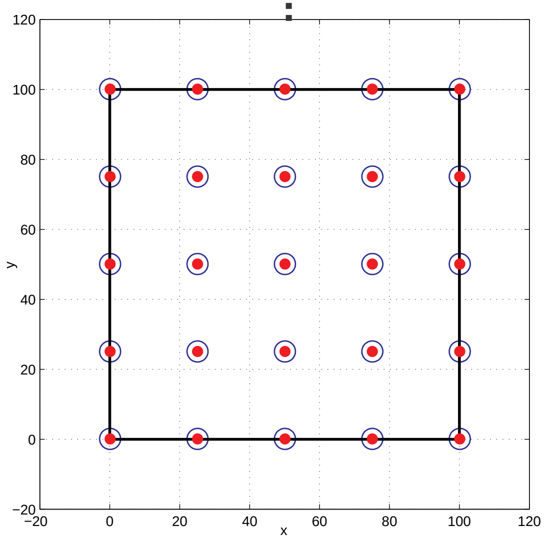
$r \rightarrow 0$

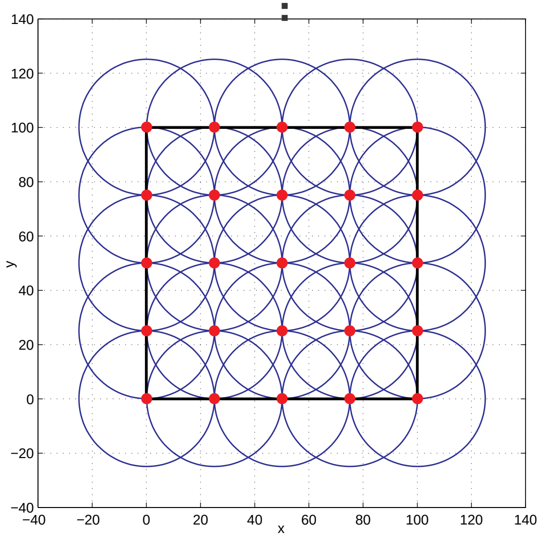
- Sensorknoten empfangen keine Beaconpositionen
- Keine Positionsbestimmung möglich (Unbekannte maximal)

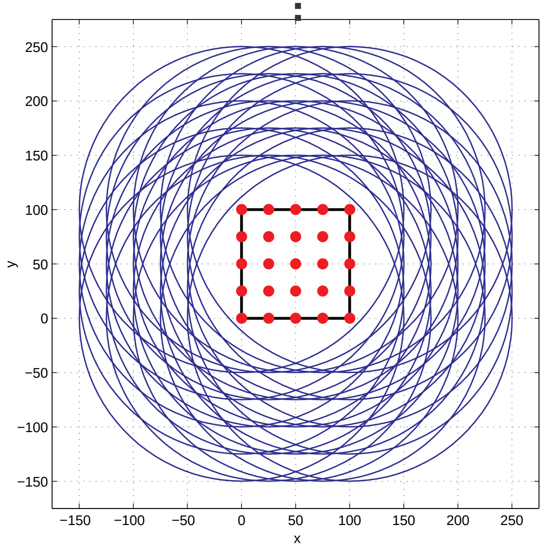
r>>Felddiagonale

- Alle Sensorknoten empfangen dieselben Beaconpositionen
- Alle Sensorknoten ermitteln gleichen Positionswert (Unbekannte = 0)

Gesucht ist die optimale Übertragungsreichweite der Beacons, bei die Anzahl der Unbekannten null und der Positionierungsfehler minimal ist.







○ : Übertragungsreichweite □ : Sensornetzwerk ● : Beacon Feldbreite: 100x100

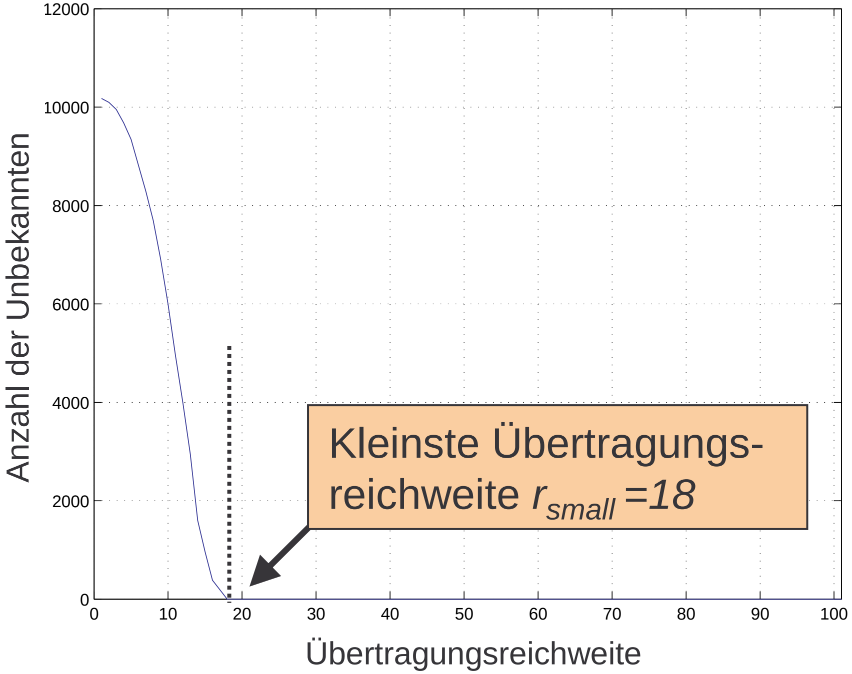
Grafische Lösung

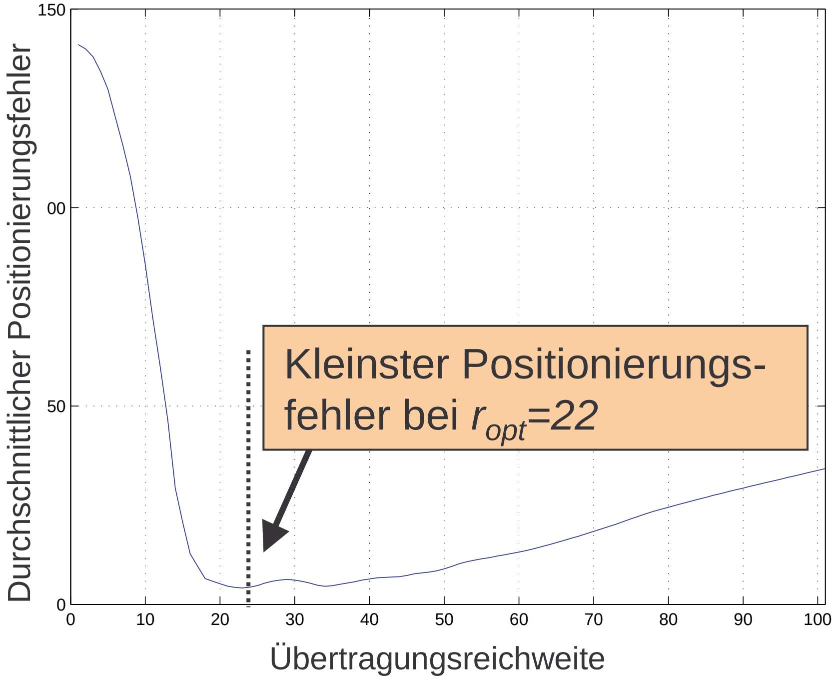
Erster Ansatz zur Minimierung des Energieverbrauchs:

Ermittlung der kleinsten Übertragungsreichweite r_{small} bei der die Anzahl der Unbekannten null ist.

Aber: Positionierungsfehler nicht minimal!

$\boxed{r_{small} < r_{opt}}$





Power-Error Produkt (PEP)

Energiebetrachtungen (ideal):

$$E_{Send} = E_{Init} + mE_{Dyn}$$

$$E_{Dyn} = E_{Bit} \cdot \left(\frac{4 \cdot \pi \cdot r}{\lambda} \right)^2$$

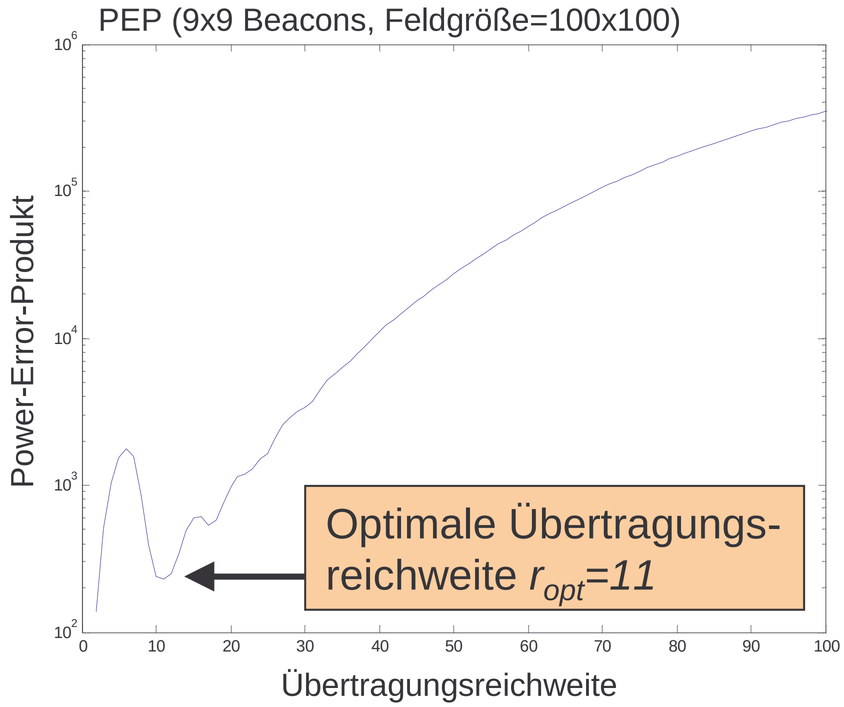
$$E_{Dyn} \Big|_{E_{Bit} \cdot \left(\frac{4 \cdot \pi}{\lambda} \right)^2 = 1} = r^2$$

m = Anzahl der zu übertragenden Bits
E_{Init} = Initialisierungsenergie des Transmitters
E_{Dyn} = Übertragungsenergie für ein Bit
λ = Wellenlänge
r = Übertragungsreichweite der Beacons
PEP = Power-Error-Produkt
L = Anzahl der Sensorknoten im Netzwerk
r = Distanz
f_i = Positionierungsfehler in einem Sensorknoten
$f_{l_{mean}}$ = Durchschnittlicher Positionierungsfehler

Power-Error-Produkt:

$$PEP = E_{Dyn} \cdot f_{l_{mean}} = r^2 \cdot \frac{\sum_{i=1}^{l} f_i}{l}$$

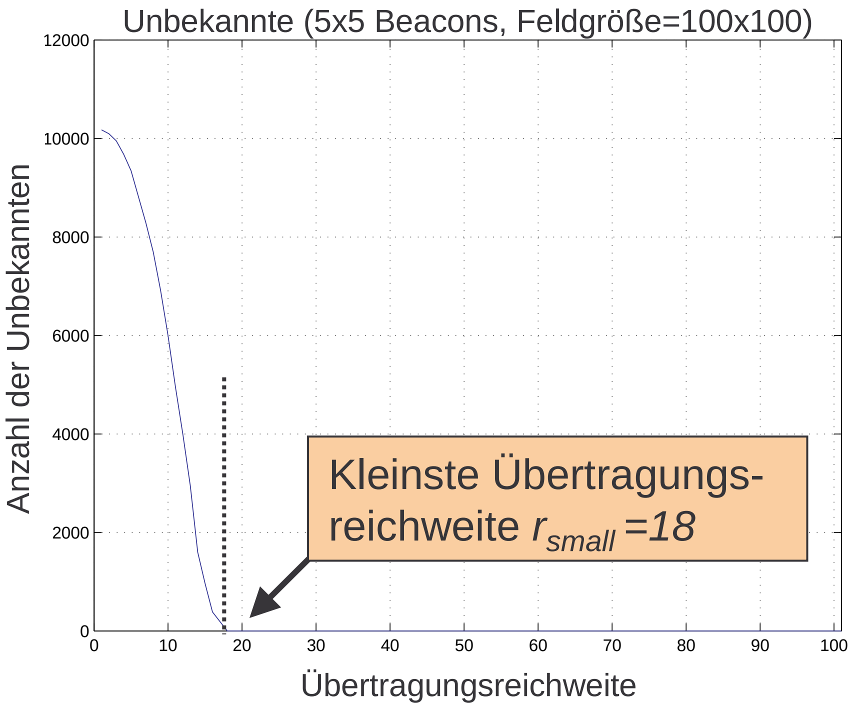
Performance-Indikator für Optimierungsproblem aus Übertragungsreichweite, Unbekannten und Positionierungsfehler

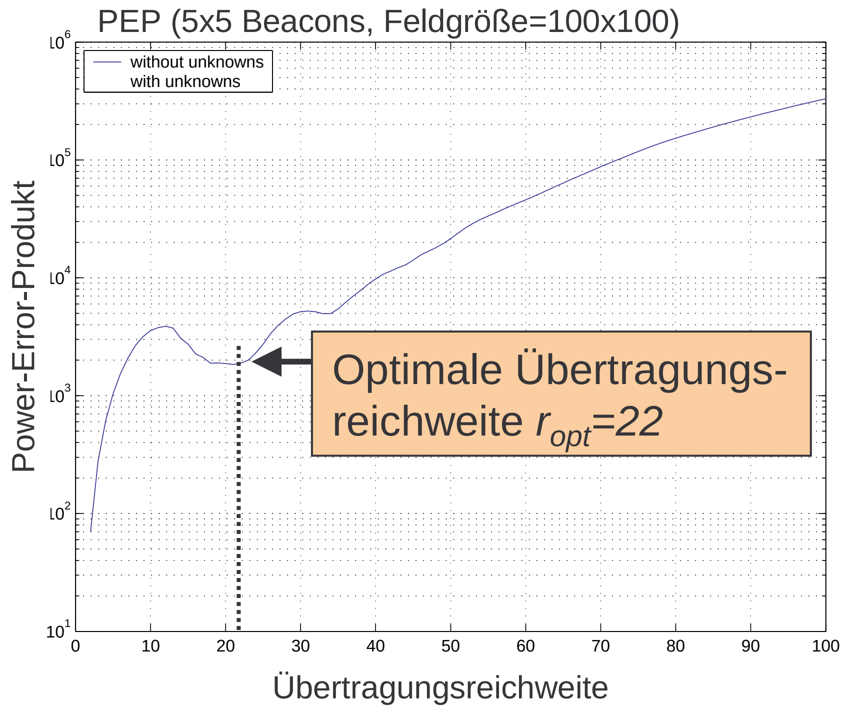


Graphical Solution II

Bestimmung der optimalen Übertragungsreichweite:

- Ermittlung von r_{small} für Unbekannte=0
- Lokalisieren des PEP-Minimums für $r_{small} \leq r_{opt}$ (hier: 18≤22)





Weitere Abhängigkeiten

Erkenntnis:

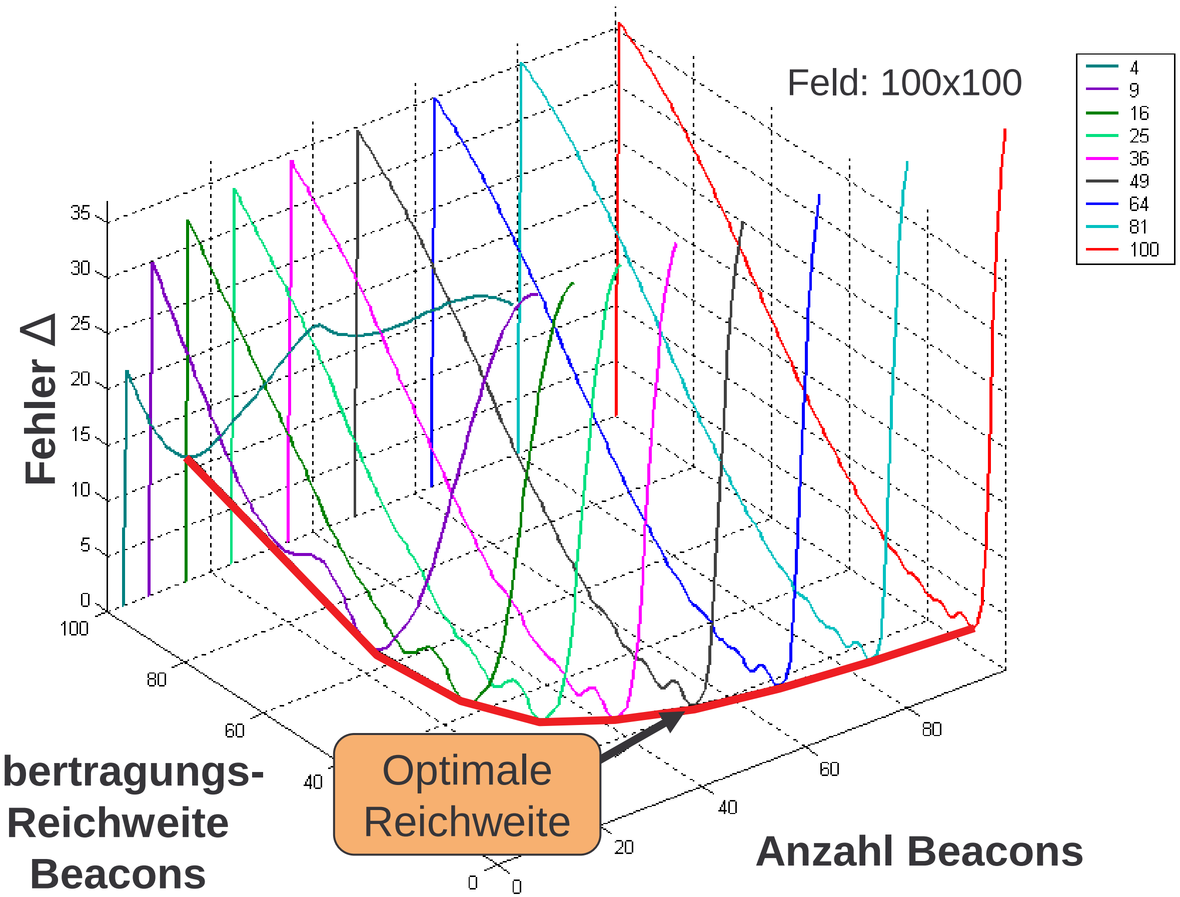
Optimale Übertragungsreichweite ist ebenfalls abhängig von der Anzahl der Beacons

Optimalkriterium

$ÜRW_{Opt} = f(\Delta, \#Beacons)$

Anwendung

Energieminimierung

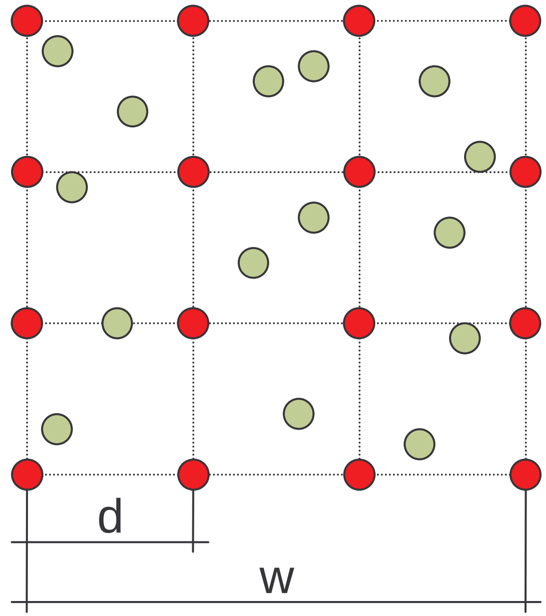


Distanz zwischen Beacons

Definitionen:

d = Distanz zwischen zwei Beacons
w = Ausdehnung des Sensornetzwerkes
n = Anzahl der Beacons

Sensornetzwerk (w=150, n=16)



Gleichung 1:

$$d = \frac{w}{\sqrt{n} - 1}$$

Beispiel:

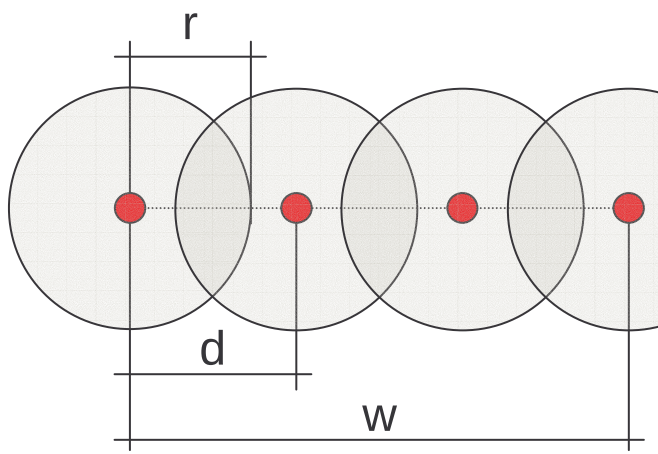
$$d = \frac{150}{\sqrt{16} - 1} = \frac{150}{3} = 50$$

: Beacon : Sensorknoten

Granularität

Granularität:

- Verhältnis zwischen Übertragungsreichweite der Beacons *r* und der Distanz *d*
- Granularität ist unabhängig von
  - Ausdehnung des Sensornetzwerkes *w*
  - Anzahl der Beacons *n*

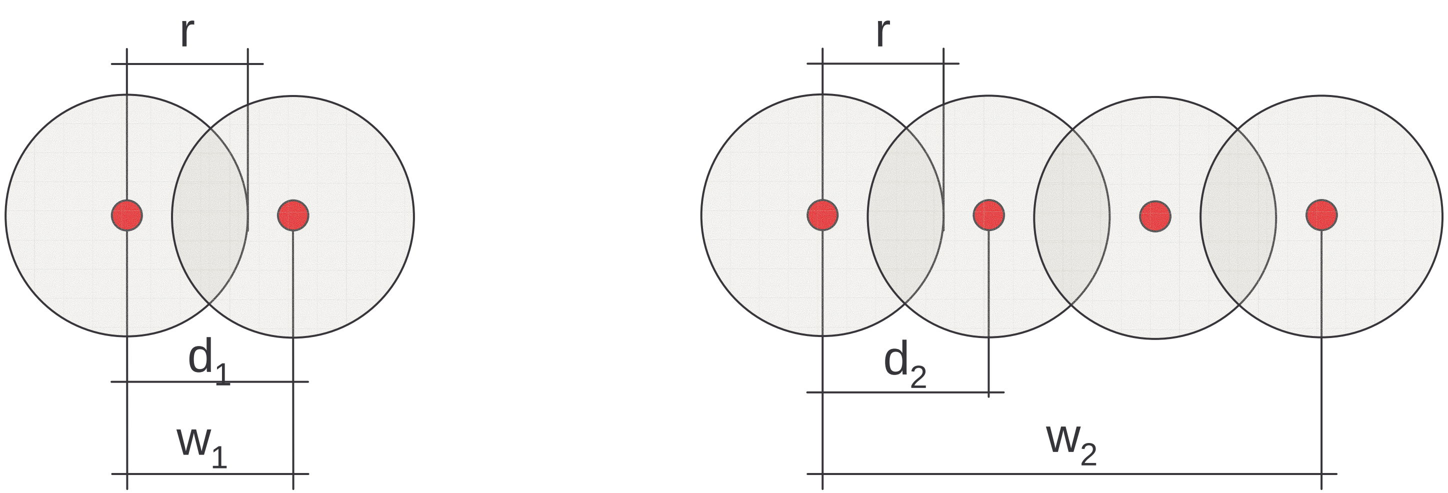


d = Distanz zwischen zwei Beacons
w = Ausdehnung
r = Übertragungsreichweite
G = Granularität
 = Übertragungsreichweite
 = Beacon

Gleichung 2: $G = \frac{r}{d}$

Skalierung

Ist die Granularität *G* wirklich unabhängig von der Feldgröße und der Anzahl der Beacons?

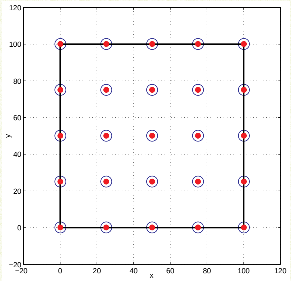
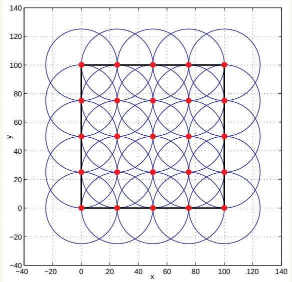
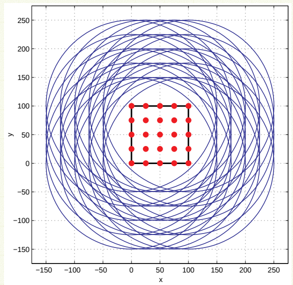


d_1 = d_2
r = r
n_1 = 4
n_2 = 16

$$G = \frac{r}{d_i}$$

G ist unabhängig von der Feldgröße *w* und der Anzahl der Beacons *n*!

Optimale Granularität

r→0	r=optimal	r>>diagonal
		
Δ = hoch G = niedrig	Δ = niedrig G = optimal	Δ = hoch G = hoch

$$G_{opt} = \frac{r_{opt}}{d} \longrightarrow \boxed{G_{opt} \textbf{ existiert!}}$$

Optimale Granularität:

- Unabhängig von Feldgröße *w* und Anzahl der Beacons *n*
- G_{opt} = konstant
- Wird benutzt, um r_{opt} zu berechnen.

Δ : Positionierungsfehler ● : Beacon ○ : Übertragungsreichweite eines Beacons

Optimale Übertragungsreichweite

Bereits eingeführt:

$$d = \frac{w}{\sqrt{n}-1}$$

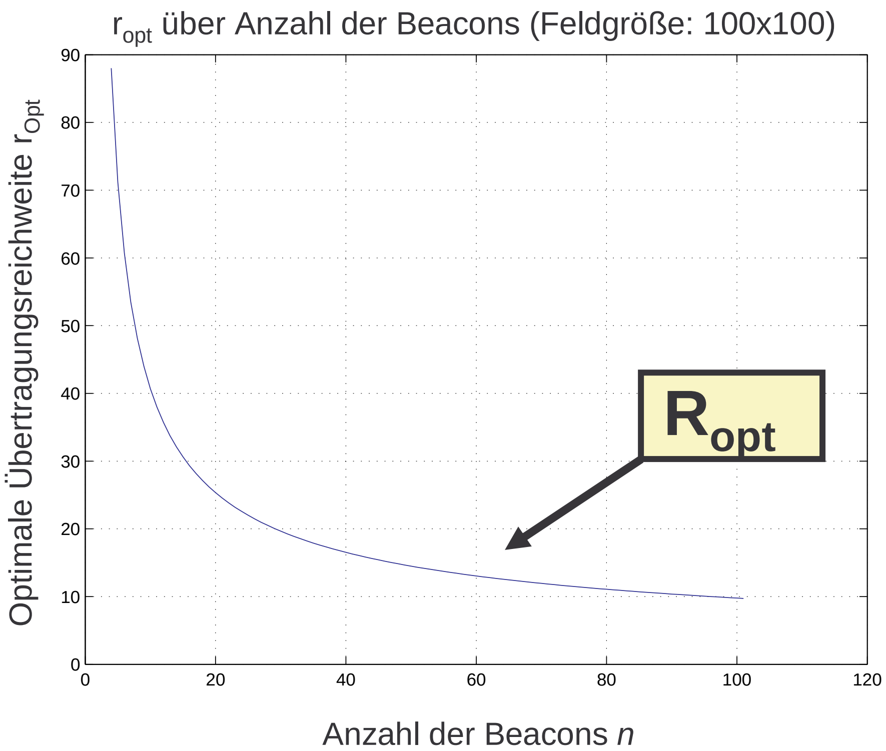
Herleitung von r_{opt}:

$$G_{opt} = \frac{r_{opt}}{d}$$

$$r_{opt} = G_{opt}\, d$$

$$r_{opt}(w,n) = \frac{G_{opt} \cdot w}{\sqrt{n}-1}$$

d = Distanz zwischen Beacons
w = Feldgröße
n = Anzahl der Beacons
r = Übertragungsreichweite
G = Granularität

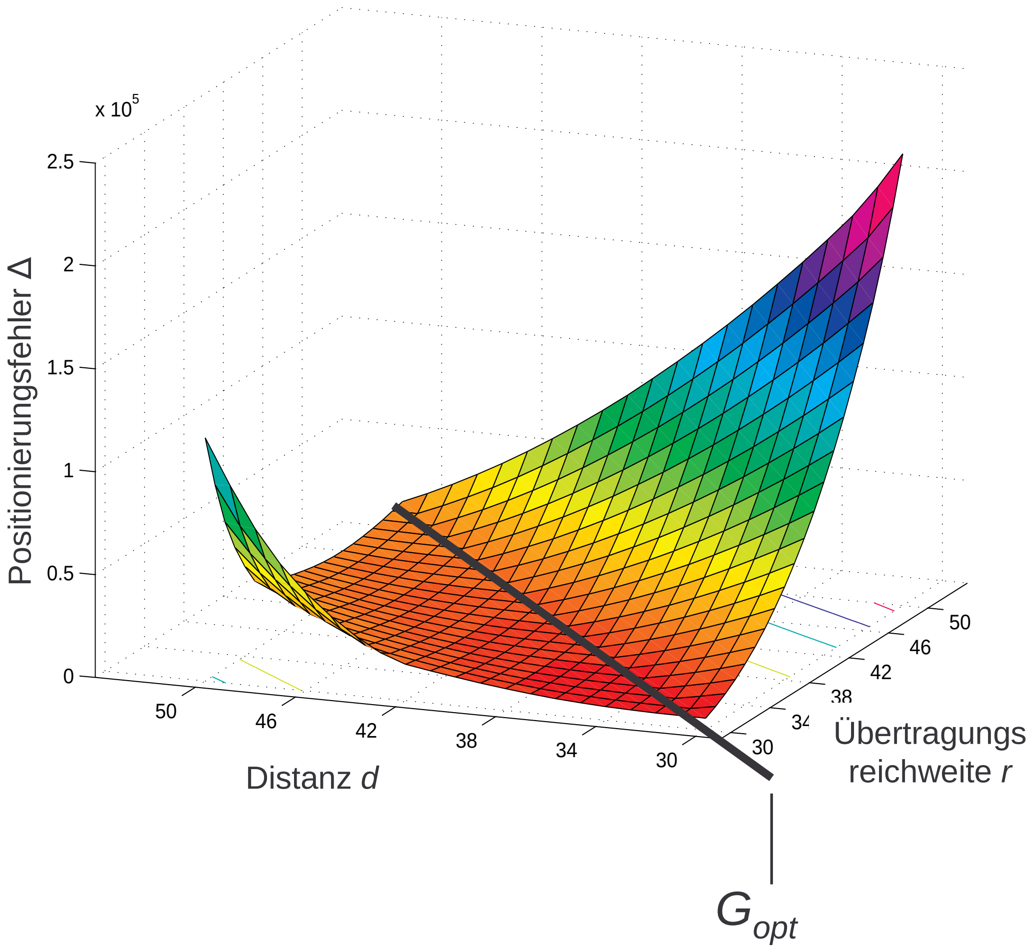


Bestimmung von G_{Opt}

Bestimmung von G_{opt} ?

- Bestimmung des Positionierungsfehlers Δ_i durch Variation von *r* und *d*
- Bestimmung von r_{opt} an Distanzen d_i mit kleinstem Positionierungsfehler

$$G_{opt_i} = \frac{r_{opt_i}}{d_i}$$

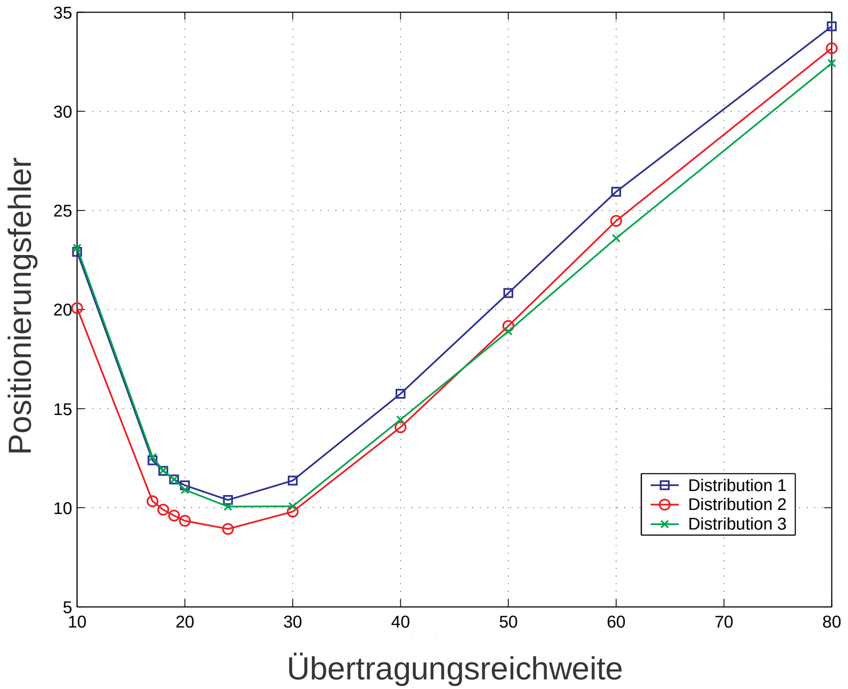


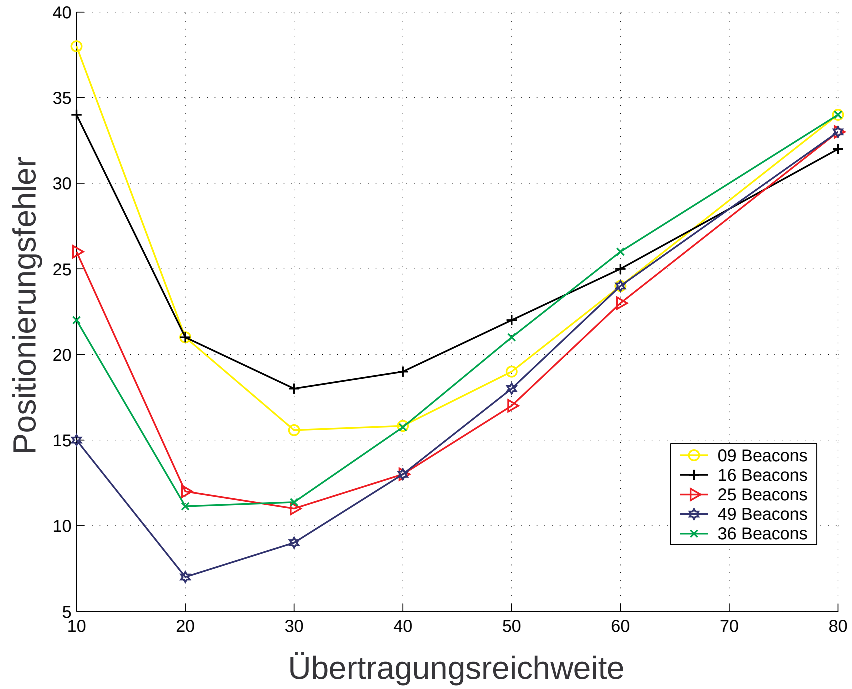
Ergebnis:

- G_{opt} is konstant !
- $G_{opt} \approx 0.86$

Stochastische Beaconverteilung

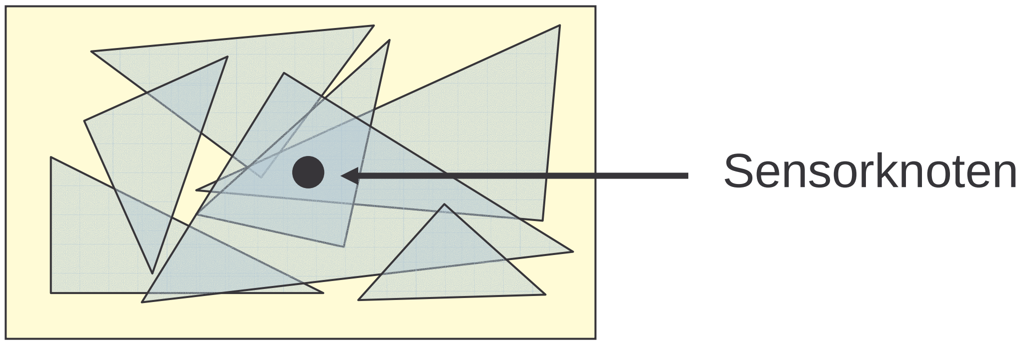
- Ähnlicher Verlauf der optimalen Übertragungs-reichweite wie im Infrastrukturfall
- Min. Positionierungsfehler steigt von 3% auf 9% gegenüber Infrastruktur-Fall





APIT: Algorithmus

- Sensornetzwerk aus Knoten und *n* Beacons
- Hohe Sensorknotendichte
- Beacons senden Positionen aus
- Sensorknoten empfangen Beaconposition
  - Messung der Empfangsfeldstärke
- Bildung von $\binom{n}{3}$ Dreiecken aus Position aller Beacons

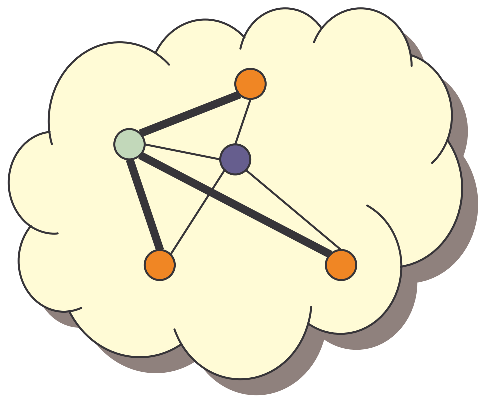


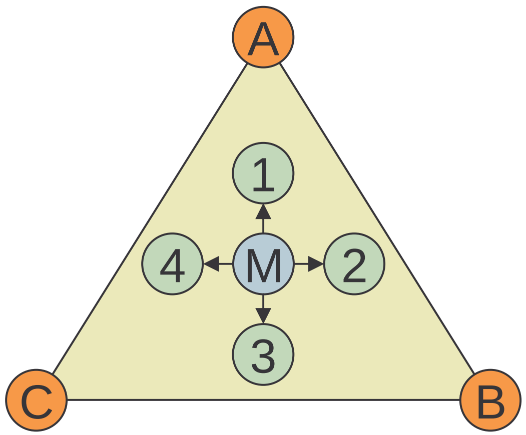
In welchen Dreiecken befindet sich der Sensorknoten ?

APIT: PIT-Test

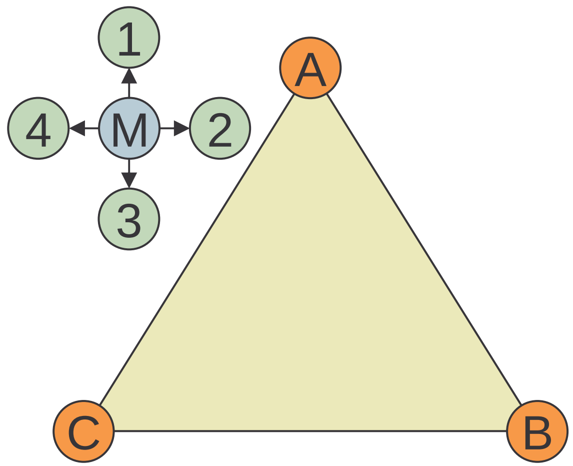
Point-In-Triangulation (PIT) Test:

- Austausch der Beacondaten von 4 Nachbarn-Sensorknoten (1,2,3,4) nötig
- Vergleich: Abstände von Nachbarn zu Beacons (A,B,C) mit eigenen Abständen zu Beacons (Bedingung: 2 kürzer, 1 länger)
- PIT bestanden, wenn Vergleich mit allen 4 Nachbarn erfolgreich





Sensorknoten im Dreieck



Sensorknoten nicht im Dreieck

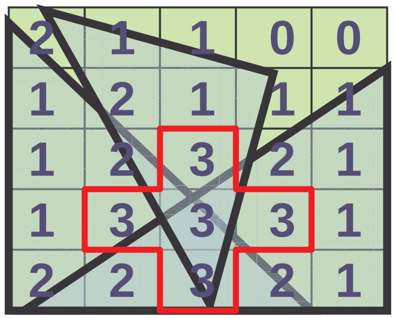
APIT: Aggregation

APIT-Aggregation:

- Diskretisierung der Dreiecke (SCAN Algorithmus)
- Überlagerung aller Dreiecke
- Übereinanderliegende Flächen inkrementieren

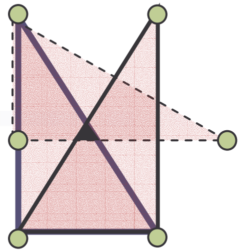
Positionsbestimmung:

- Schwerpunktbildung aus resultierender Fläche



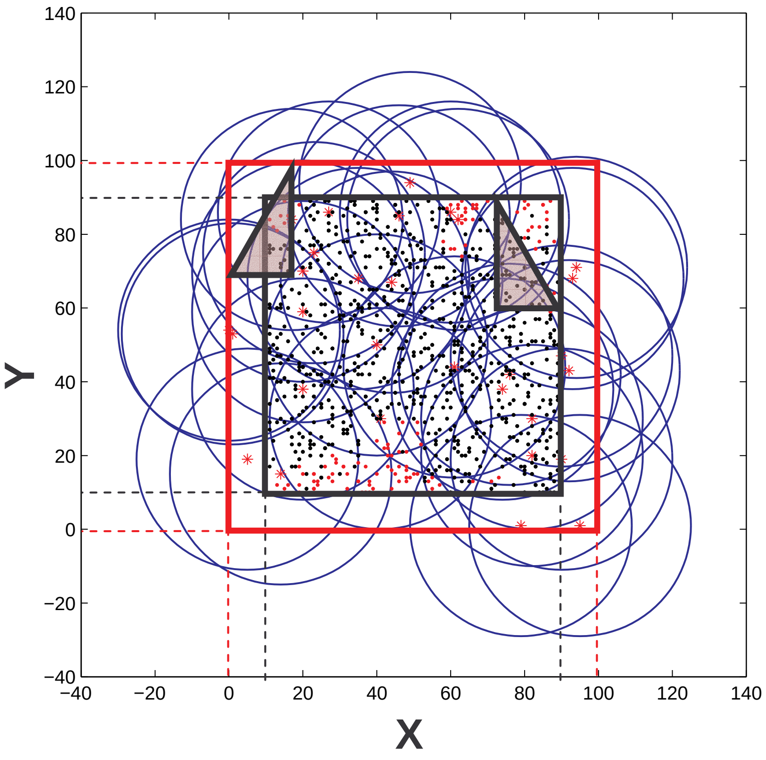
APIT-Aggregation

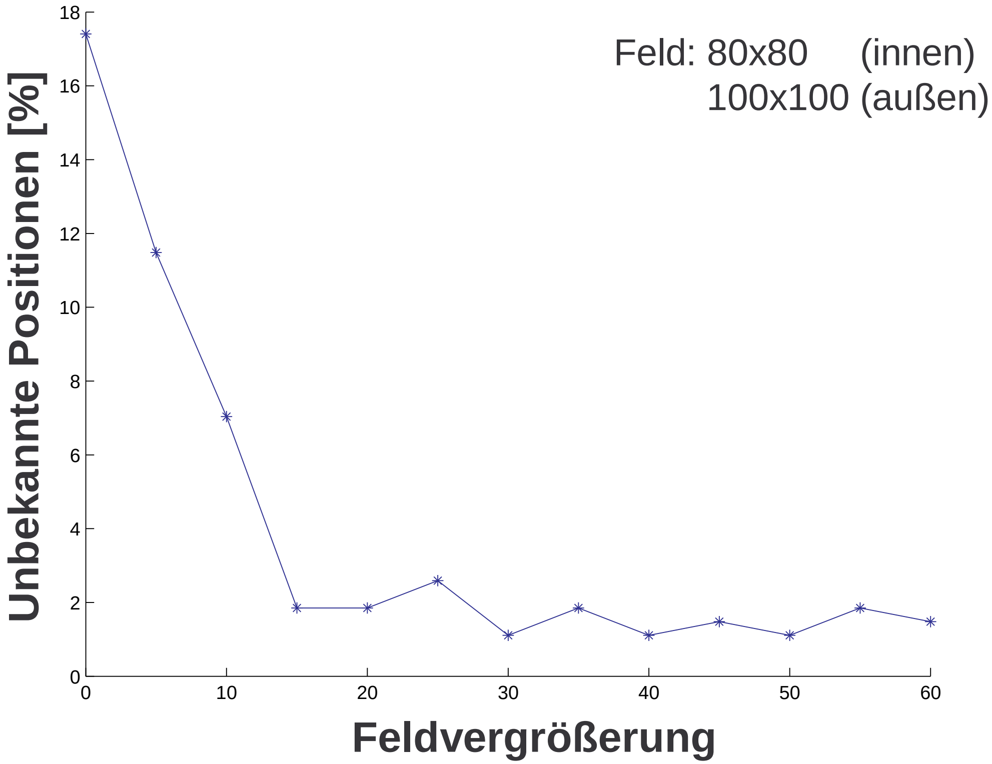
Verbesserung APIT



Anzahl positionsloser Knoten verringern:

- Ansatz: **Weitflächigere Verteilung der Beacons**
  Bestimmung eines optimalen äußeren Bereiches
- Vorteil: **< 2% unbekannte Positionen**
- Nachteil: Steigender Fehler durch größere Dreiecke



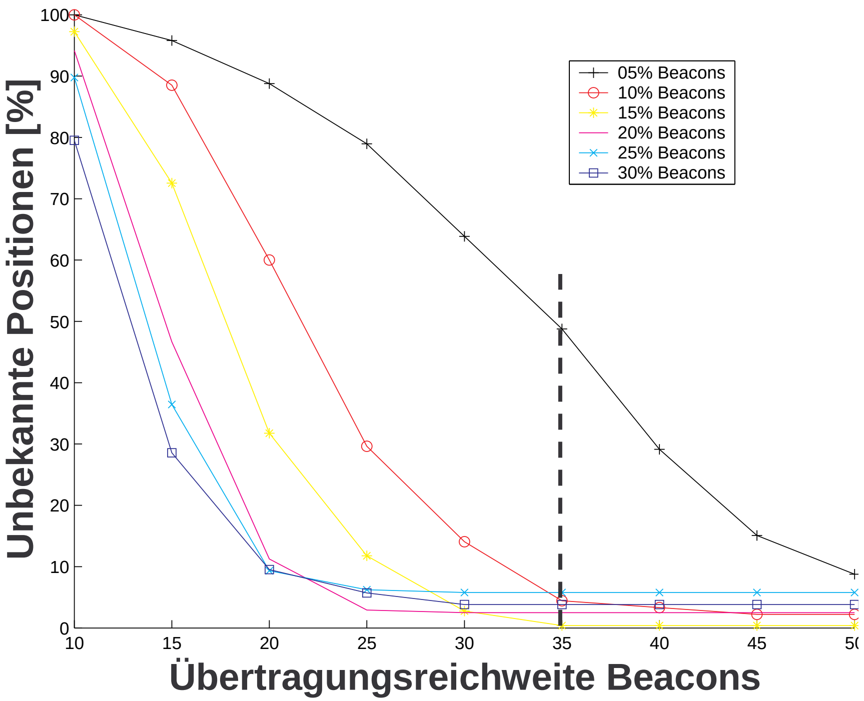


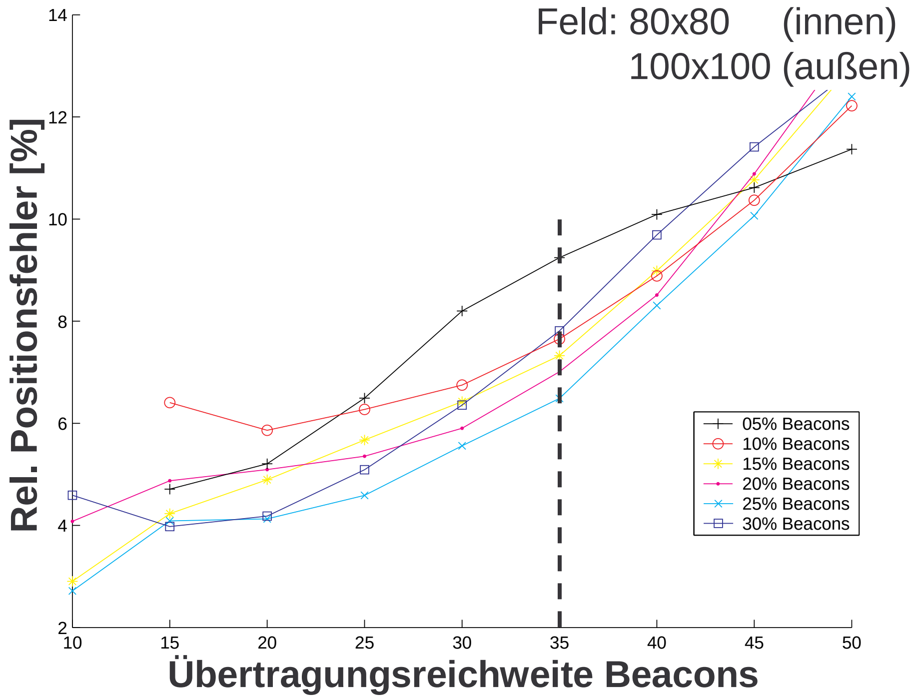
Verbesserung II - APIT

Frage: **Optimale Übertragungsreichweite** der Beacons

Ergebnis: Kompromiss: Reichweite Beacons ⇔ # unbekannte Positionen ⇔ Positionierungsfehler Δ

Beispiel: Gegeben: Anzahl der Beacons 10%, ≤5% unbekannte Positionen
Lösung: Beacon-Reichweite$_{optimal}$ = 35, Rel. Δ = 7.5%





Verringerung des Energieaufwands für Kommunikation und Berechnung

Literatur

Coarse Grained

N. Bulusu, et. al.: *„GPS-less Low Cost Outdoor Localization For Very Small Devices“*

J. Blumenthal, F. Reichenbach, M. Handy, D. Timmermann: *„Optimal Adjustment Of The Coarse Grained Localization-Algorithm For Wireless Sensor Networks“,* Proceedings of 1st Intl. Workshop on Positioning, Navigation, and Communication WPNC'2004, Hanover, March 2004

APIT

Tian He, et. al.: *„Range-Free Localization Schemes for Large Scale Sensor Networks“, MobiCom 2003*

Vielen Dank