Localization: Klassifikation

Motivation

In vielen Fällen bekommen die Messwerte erst dadurch ihren Wert, dass die Position bekannt ist, an der er erfaßt wurde. Daher ist die Ortsbestimmung für Sensorknoten in den meisten Fällen wichtig, jedoch nicht trivial.

Grundsätzlich könnte die Ortsbestimmung über GPS (Global Positioning System) erfolgen, wobei relativ zum Knoten selbst teure Hardware nötig ist (Preis für ein GPS Modul heute in der Größenordnung von 100 EUR). Darüber hinaus ist die Hardware zu groß und schwer und wird zudem noch einmal am Anfang gebraucht, bei stationären Knoten danach nicht mehr.

Localization: Klassifikation

Einteilung in Klassifikations-Schemata

Es gibt zwei Kombinierbare Dimensionen, denn **Knoten kennen ihre Position** oder nicht und **Knoten können die Entfernung benachbarter Sender bestimmen** oder nicht.

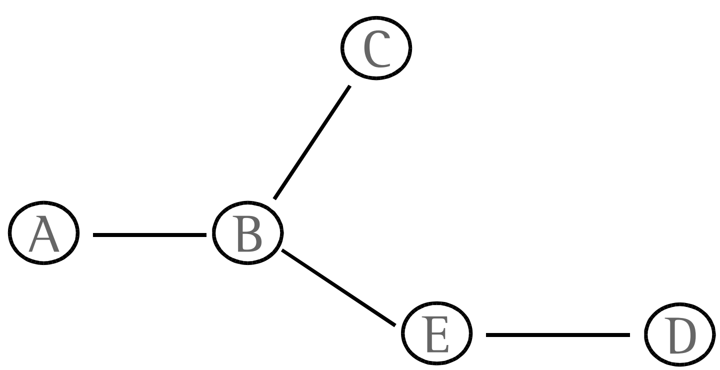
Entfernung benachbarter Sender abschätzbar \ Einige Knoten kennen ihre Position	ja	nein
ja	Vollst. Topologie in Weltkoordinaten berechenbar	Vollst. Topologie berechenbar, Koordinatensystem jedoch nur lokal
nein	Grobe Positionierung innerhalb der Nachbarschaft	Nur abstrakte Matrix der Nachbarschaften berechenbar

Localization: Unbekannte Positionen und Entfernungen

Fall: Keine Knotenposition, keine Entfernungsabschätzung

Die einzelnen Knoten kennen ihre globale Position nicht. Auch können Sie die Entfernung der Nachbarn nicht abschätzen, die gehört werden können. Hier ist nur das Austellen einer Adjazenzmatrix möglich.

	Knoten A	Knoten B	Knoten C	Knoten D	Knoten E
Knoten A	1	1			
Knoten B	1	1	1		1
Knoten C		1	1		
Knoten D				1	1
Knoten E		1		1	1

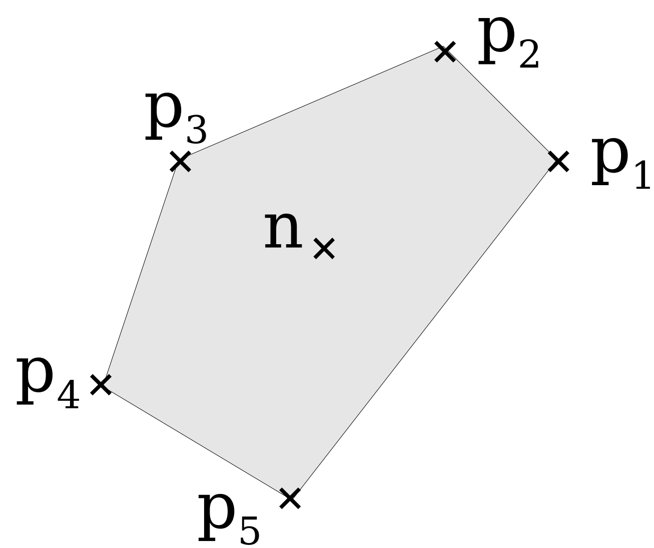


Localization: Bekannte Positionen ohne Entfernungen

Bildung eines gleichgewichteten Mittels

Bilde das Mittel der Koordinaten der Nachbarknoten innerhalb der Reichweite des Empfängers n. Bemerkung: Spannen die Sender ein konvexes Polygon auf, so ist das Mittel der Koordinaten der Schwerpunkt des Polygons.

$$n = \frac{1}{|P|} \sum_{\forall p_i \in P} p_i$$



Localization: Bekannte Positionen und Entfernungen

Fall: Es gibt Knoten, deren lokale Position bekannt ist

Embedded System in Autos oder Handies und PDAs sind auch Knoten eines Netzwerkes, auch wenn sie vornehmlich anderen Aufgaben dienen). Sie verfügen oft über bessere Energiequelle oder werden regelmäßig aufgeladen.

Idee für lokal feste Knoten: Sensornetze nutzen diese Embedded Systems als Location-Server. Andere Möglichkeit: Mit der großen Anzahl billiger Sensoren werden einige teure verteilt, die auch GPS-fähig sind. Die einfachen Sensoren sollen sich dann über die aufwändigen lokalisieren. Die GPS-fähigen Knoten müssen dabei nur kurze Zeit lebensfähig bzw. in der Nachbarschaft verfügbar sein – zumindest so lange, bis die fixen Knoten Ihre Position ermittelt haben.

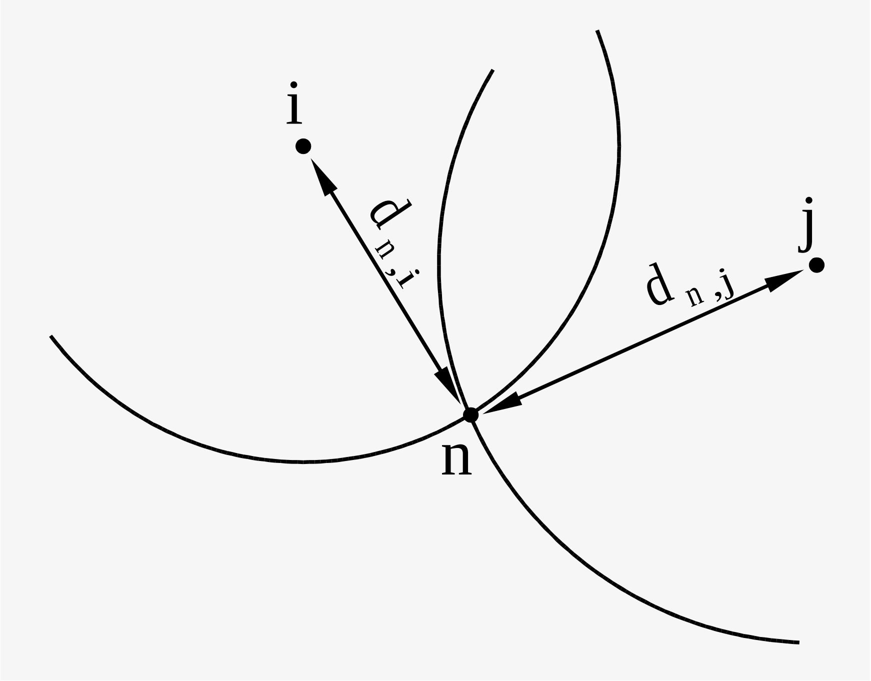
Alternativ könnten einige der einfachen Knoten von Hand ausgesetzt und z. B. per GPS vermessen werden. Diese dienen den anderen dann als Location Server, jedoch GPS-less auf Kosten eines höheren Initialisierungsaufwands.

Für bewegliche Knoten (z. B. an Tieren, im Wasser etc.): Wie in der Nautik üblich könnten feste, öffentliche Funkfeuer benutzt werden.

Localization: Bekannte Positionen und Entfernungen

Fall: Es gibt Knoten, deren lokale Position bekannt ist

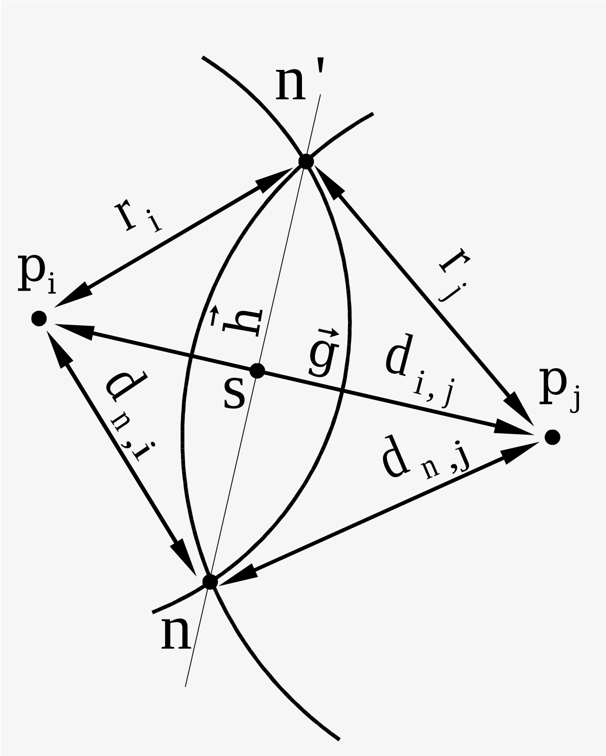
Grundsätzlich reichen drei Knoten aus, um die Position eines neuen Knoten zu berechnen, setzt man voraus, dass die Abstände paarweise bekannt sind. Die Abstände der Knoten untereinander können über die Feldstärke abgeschätzt werden, die der Empfängerknoten detektiert (analog im Raum?).

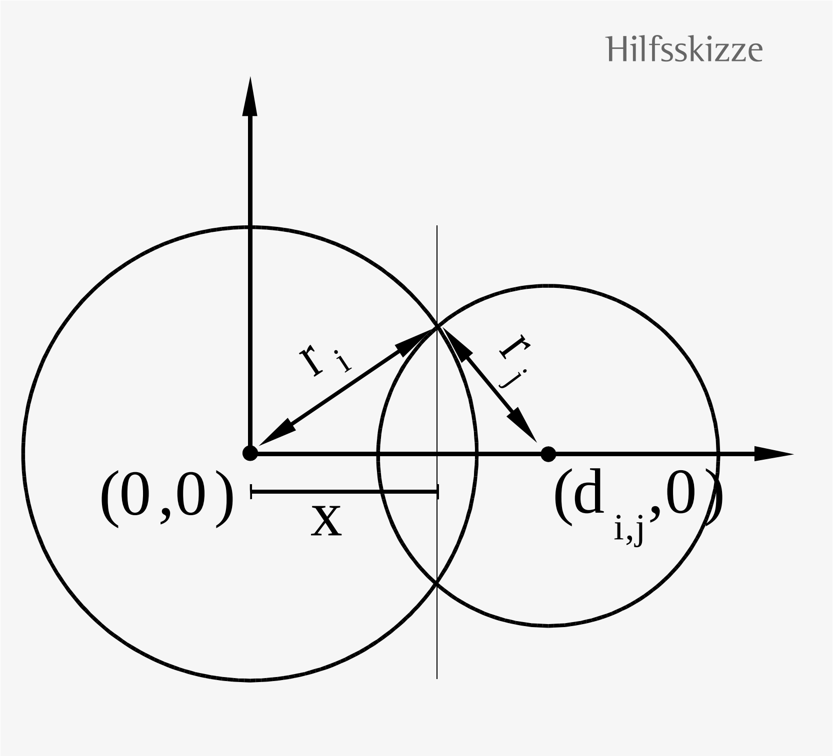


Localization: Bekannte Positionen und Entfernungen

Fall: Es gibt Knoten, deren lokale Position bekannt ist

Gegeben sind die Knoten i und j auf den Positionen p_i und p_j. Der Radius um p_i durch den gesuchten Knoten n entspricht dem Abstand $d_{i,n}$ (analog für Knoten j). Um zu n zu gelangen werden die Positionen s und die Distanz $d_{s,n}$ so wie das Lot h auf g benötigt ($d_{i,j}$ entspricht gerade der Länge von g). Bemerkung: Um sich letztlich auf einen der beiden Knoten festlegen zu können, muss ein dritter Nachbar hinzugezogen werden.





Localization: Bekannte Positionen und Entfernungen

Fall: Es gibt Knoten, deren lokale Position bekannt ist

Punkte auf dem Kreis um (0,0):

$$x^2 + y^2 = r_i^2$$

Punkte auf dem Kreis um (d,0):

$$(x - d_{i,j})^2 + y^2 = r_j^2$$

Berechnen der gemeinsamen x-Koordinate durch Einsetzen:

$$(x - d_{i,j})^2 + (r_i^2 - x^2) = r_j^2$$

$$x^2 - 2\, d_{i,j}\, x + d_{i,j}^2 + r_i^2 - x^2 = r_j^2$$

$$x = \frac{d_{i,j}^2 - r_j^2 + r_i^2}{2\, d_{i,j}}$$

Localization: Bekannte Positionen und Entfernungen

Fall: Es gibt Knoten, deren lokale Position bekannt ist

p_i und p_j bezeichnen die Koordinaten der Knoten i und j. Der Punkt s im linsenförmigen Schnittgebilde der beiden Kreise ergibt sich durch:

$$s = p_i + x(p_j - p_i)$$

Das Lot h auf g ist im Zweidimensionalen bis auf das Vorzeichen eindeutig bestimmbar:

$$\vec{h} = \pm \begin{pmatrix} -g.y \\ +g.x \end{pmatrix}$$ bzw. das auf die Länge 1 normierte Lot $$\vec{h}^{*} = \frac{\pm 1}{|\vec{h}|}\vec{h}$$

Probe:

$$\begin{pmatrix} g.x \\ g.y \end{pmatrix}\vec{h} = \begin{pmatrix} g.x \\ g.y \end{pmatrix} \pm \begin{pmatrix} -g.y \\ +g.x \end{pmatrix} = \pm(-g.x \times g.y + g.y \times g.x) = 0$$

Localization: Bekannte Positionen und Entfernungen

Fall: Es gibt Knoten, deren lokale Position bekannt ist

Aus dem Produkt d_{sn} h^* ergeben sich zusammen mit dem Punkt s nun zwei Kandidaten n und n'. Der Abstand d_{sn} zwischen s und n ergibt sich aus dem rechtwinkligen Dreieck zwischen p_i, s und n.

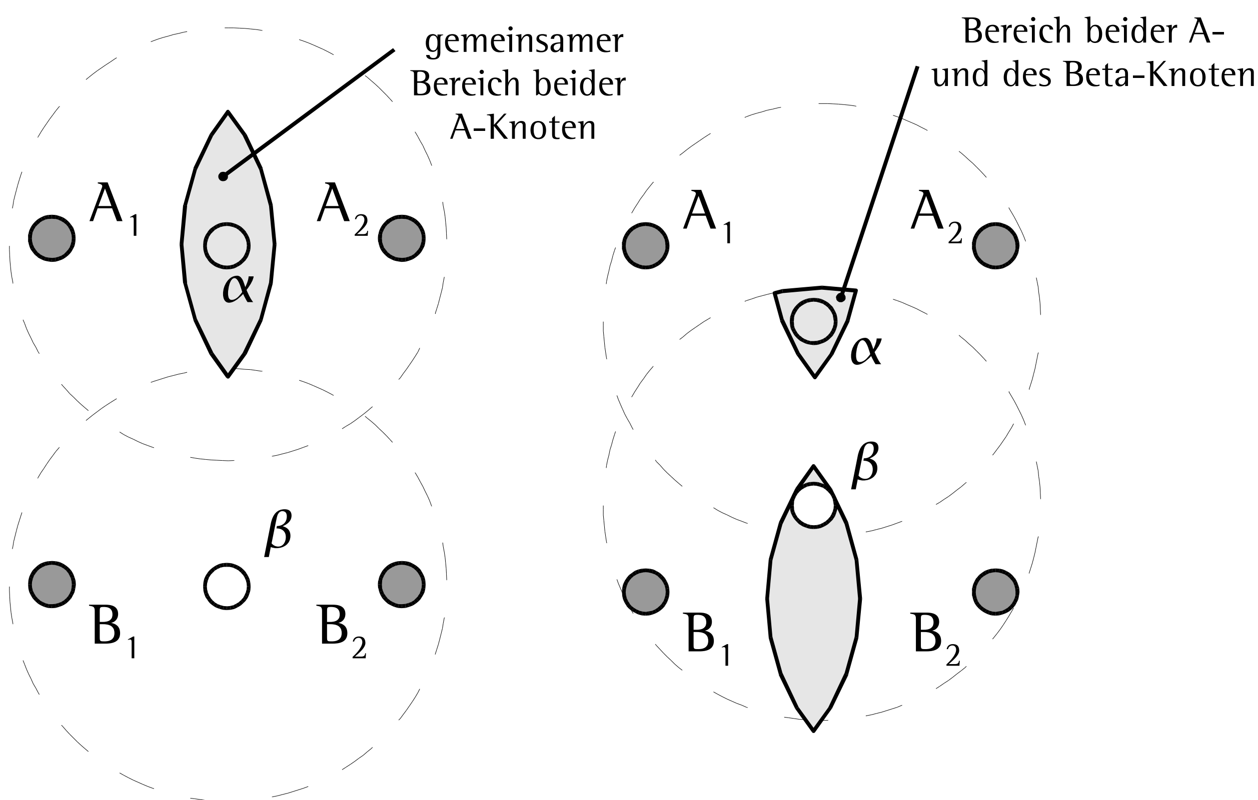
Im letzten Schritt muss herausgefunden werden, welcher der Fälle n oder n' vorliegt. Dies ist in der Regel durch einen einfachen Vergleich mit einem weiteren Knoten in der Nähe zu leisten, alleine durch Überprüfen der Nachbarschaftsbeziehung. Unter den beiden Varianten für n kann nur diejenige die richtige sein, für die auch alle Nachbarn in Empfangsreichweite liegen.

Im Dreidimensionalen funktioniert die Ermittlung analog, jedoch mit Hilfe von drei global lokalisierbaren Knoten. Eine Fallunterscheidung erübrigt sich zumindest in der satellitengestützten Lokalisierung, da ein Punkt auf dem Erdmantel liegt und sich der andere weit außerhalb im Weltall befindet.

Localization: Bekannte Positionen ohne Entfernungen

Zusätzliche Berücksichtigung der unlokalisierten Nachbarschaft

Auf den ersten Blick ist nur die Berücksichtigung der Nachbarn mit Positionsdaten nützlich. Bei genauerer Analyse zeigt sich, dass auch die Erreichbarkeit nicht exakt positionierter Nachbarn den eigenen „unsicheren" Bereich einschränken können. Wenn sich Knoten beta am nördlichen Ende seines möglichen Bereichs befindet, schränkt es den Bereich von Knoten alfa immer noch (kleinstmöglich) ein, obwohl die wahre Position von beta nicht bekannt ist.



Localization: Bekannte Positionen ohne Entfernungen

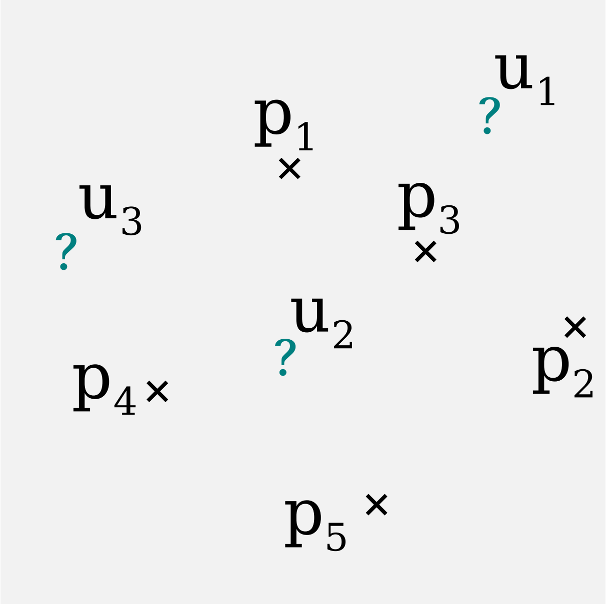
Zusätzliche Berücksichtigung der unlokalisierten Nachbarschaft

Das Problem der Positionierung von Knoten ohne diese Information kann wie folgt als Optimierungsproblem formuliert werden.

$$\arg\min_{\forall (u_1, u_2, \ldots, u_k)} \sum_{\forall d_{i,k}} \left| \|u_i - x_k\|_2 - d_{i,k} \right|$$

u_i bezeichnet die Position eines Knoten i, p_j die Position eines lokalisierten Knoten j. Als x wird ein Knoten aus der Vereinigungsmenge beider Knoten bezeichniet.

$d_{i,j}$ beschreibt die Distanz zwischen zwei Knoten i und j, die z. B. aus der Feldstärke des empfangenen Signales abgeleitet werden kann.



Localization: Bekannte Entfernungen ohne Positionen

Motivation

Der folgende Ansatz aus der Veröffentlichung von Capkun, Hamdi und Hubaux

„GPS-free positioning in mobile ad-hoc networks“

baut auf Grundlage von gegenseitigen Distanzmessungen ein globales Koordinatensystem (KS) des Netzwerkes auf. Letztlich wird dabei ein zentraler Knoten als Ursprung gewählt und alle anderen Knoten in dessen KS einbezogen.

Hierzu muss zuerst jeder Knoten ein eigenes lokales Koordinatensystem erzeugen, in das nur seine unmittelbaren Nachbarn einbezogen werden. Danach folgt die Konsolidierung der vielen lokalen KS zu einem einzigen globalen. Wo dieses KS in der realen Welt liegt und wie es gedreht ist, kann so nicht ermittelt werden, da alle Berechungen auf Grundlade gegenseitiger Entfernungen „selbstbezogen“ sind und kein Verankerung mit Weltkoordinaten existiert.

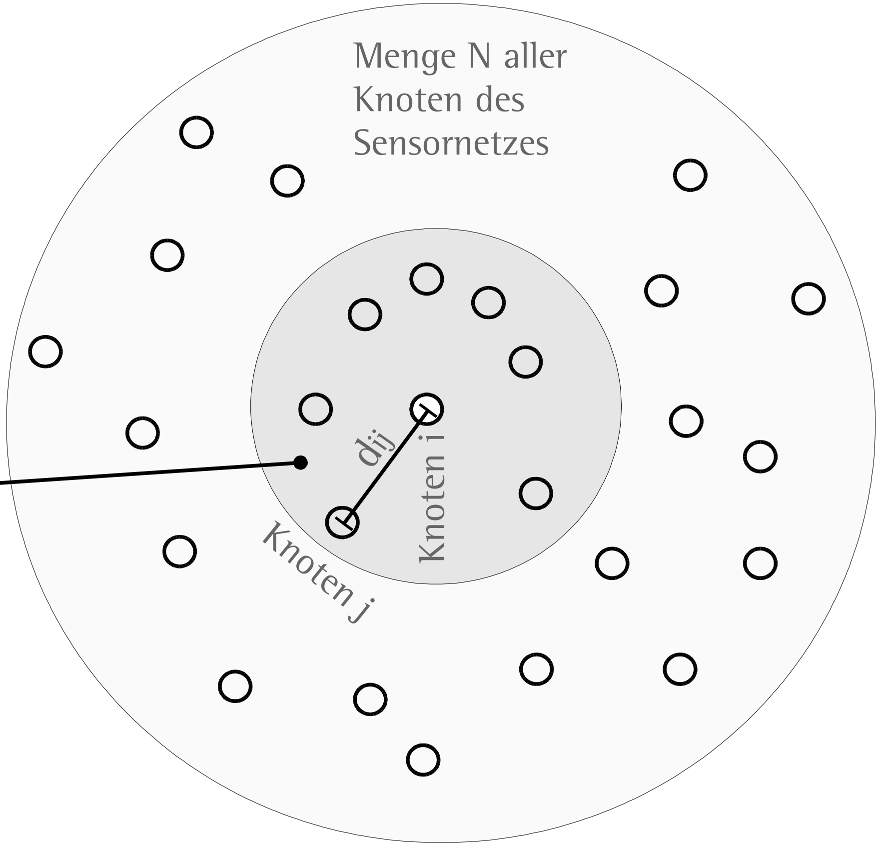
Ist das globale (und relative) KS gedoch ermittelt, so reichen bereits zwei global lokalisierte Knoten aus, um alle anderen in Weltkoordinaten interpretieren zu können.

Localization: Bekannte Entfernungen ohne Positionen

Voraussetzung

Distanzen zwischen den Knoten können abgeleitet werden aus der...

(a) Stärke des empfangenen Signals
(b) Der Round-Trip Time eines Beacons
(c) durch Versenden akkustischer Signale



Local View Set (LVS): Menge K_i von Nachbarknoten des Knoten i (oder 1-Hop Nachbarschaft)

d_{ij} = Distanz zwischen Knoten i und j

Localization: Bekannte Entfernungen ohne Positionen

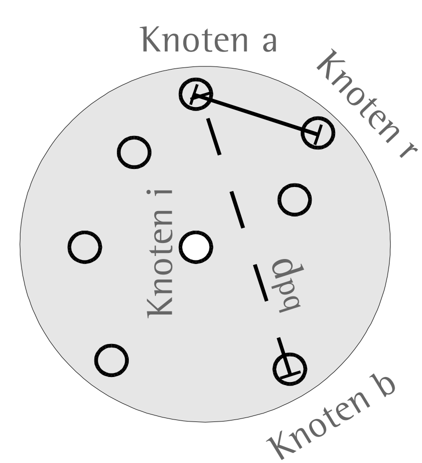
Bestimmung eines lokalen Koordinatensystems (1)

(1) Jeder Knoten bestimmt seine lokalen Nachbarknoten, also die 1-Hop Nachbarschaft. Zwischen ihnen besteht direkte Funkverbindung.

(2) Jeder Knoten verschickt die Distanzvektor Tabelle aus (1) an alle seine Nachbarn. Jeder Knoten kennt...

- seine 1-Hop und 2-Hop Nachbarschaft
- die Entfernung der direkten Nachbarn
- einige Entfernungen seiner direkten Nachbarn untereinander, jedoch nicht alle!

Knoten a kennt z. B. seinen Nachbarn r. Dies teilt er auch Knoten i mit. Knoten a und b kennen sich jedoch nicht gegenseitig. Daher können sie auch Knoten i ihre Distanz nicht mitteilen. Folge: i kennt nur einige Distanzen seiner direkten Nachbarn untereinander.



Localization: Bekannte Entfernungen ohne Positionen

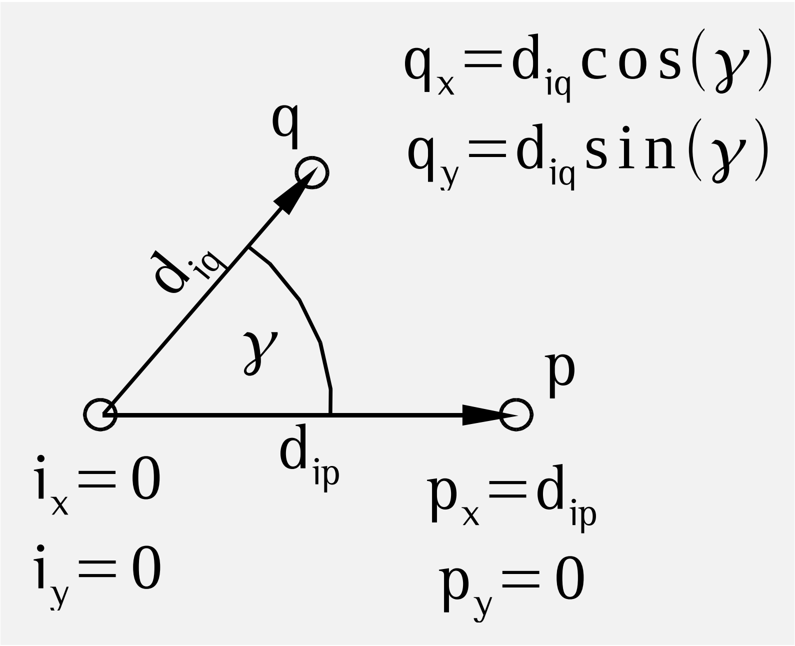
Bestimmung eines lokalen Koordinatensystems (2)

(3) Jeder Knoten bestimmt anfangs ein lokales Koordinatensystem so, dass er selbst in dessen Urspring sitzt.

(4) Ein Konten p aus der Menge LVSi wird gewählt, der die Richtung der X-Achse definiert. Dabei spielt die tatsächliche Lage des Knoten (in Weltkoordinaten) keine Rolle. Per Definition liegt der Knoten p nun auf der X-Achse in der Koordinaten (d_{ip},0).

Ein weiterer Konten q wird gewählt, der Hilft die Y-Achse zu definieren. Da die Y-Achse in der Ebene bereits als das Lot auf die X-Achse festgelegt ist, dient q lediglich dazu die Richtung der Achse zu bestimmen.

Um q tatsächlich in Koordinaten angeben zu können fehlt noch ein Winkel Gamma.



Localization: Bekannte Entfernungen ohne Positionen

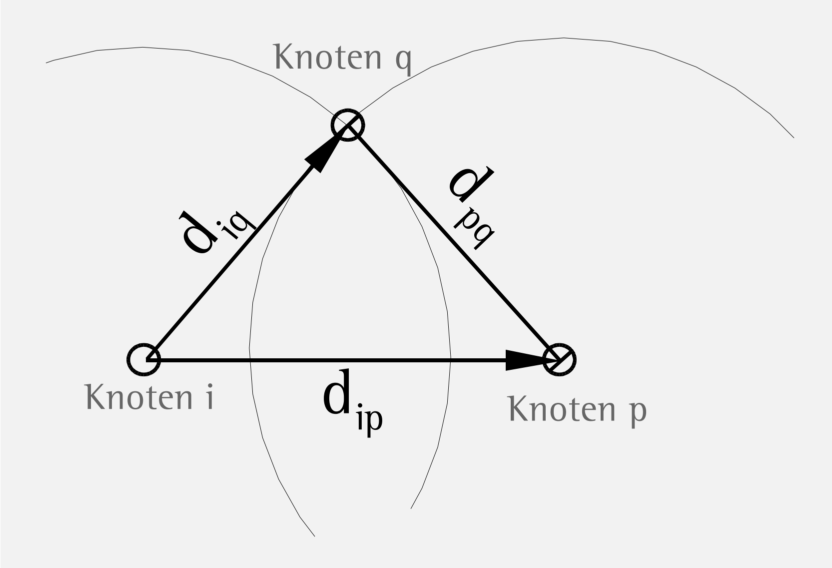
Bestimmung eines lokalen Koordinatensystems (3)

Knoten q zeichnerisch ermitteln:

Er ergibt sich aus dem Schnitt der beiden Kreise

um den Knoten i mit Radius d_{iq} und
um den Knoten p mit Radius d_{pq}.

Zu Berechnung der Koordinaten von Knoten q genügt der Winkel Gamma (siehe vorherige Seite) und der Abstand d_{iq}. Dieser kann aus den sog. Cosinus-Regeln abgeleitet werden.

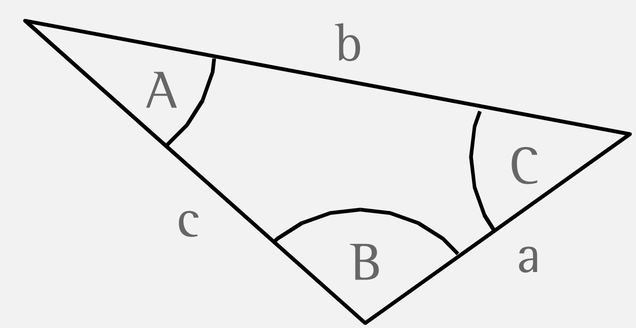


Localization: Geometrische Zusammenhänge

Cosinus Regel - Erinnerungsfolie

Gegeben ist jeweils ein Winkel und die Länge der beiden beteiligten Schenkel eines Dreiecks. Daraus läßt sich die Länge der Seite berechnen, die dem Scheitel gegenüber liegt.

$$a^2 = b^2 + c^2 - 2bc \times \cos(A)$$
$$b^2 = a^2 + c^2 - 2ac \times \cos(B)$$
$$c^2 = a^2 + b^2 - 2ab \times \cos(C)$$



Beispiel:

$$a = 4{,}3$$
$$c = 6$$
$$B = 103°$$
$$b = ?$$

$$b^2 = 4{,}3^2 + 6^2 - 2 \times 4{,}3 \times 6 \times \cos(103)$$
$$b = \sqrt{4{,}3^2 + 6^2 - 2 \times 4{,}3 \times 6 \times \cos(103)}$$

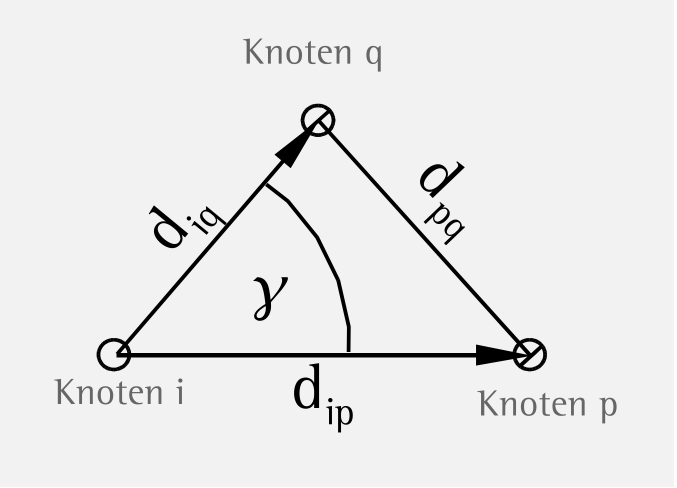
Localization: Bekannte Entfernungen ohne Positionen

Bestimmung eines lokalen Koordinatensystems (3)

$$d_{pq}^2 = d_{ip}^2 + d_{iq}^2 - 2\,d_{ip}\,d_{iq}\cos(\gamma)$$

$$2\,d_{ip}\,d_{iq}\cos(\gamma) = d_{ip}^2 + d_{iq}^2 - d_{pq}^2$$

$$\gamma = \arccos\left(\frac{d_{ip}^2 + d_{iq}^2 - d_{pq}^2}{2\,d_{ip}\,d_{iq}}\right)$$



Localization: Bekannte Entfernungen ohne Positionen

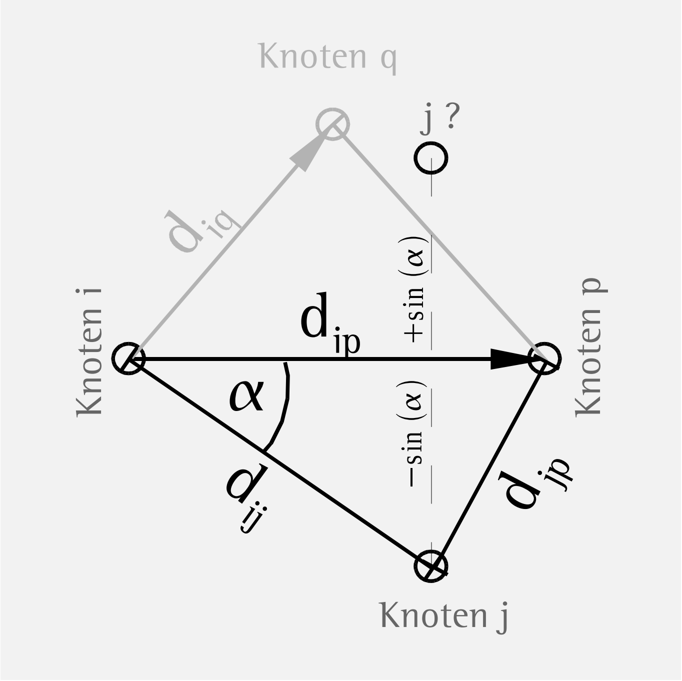
Bestimmung eines lokalen Koordinatensystems (3): Einbetten der sichtbaren Knoten

Um einen weiteren Knoten j zu bestimmen, wird neben dem Knoten i scheinbar zuerst nur der Knoten p benötigt. Allerdings machen die Abstände der Knoten untereinander keine Aussage darüber, auf welcher Seite sich Knoten j befindet.

$$j_x = d_{ij} \cos(\alpha)$$

$$j_y = (\pm 1) d_{ij} \sin(\alpha)$$

$$\alpha = \arccos\left(\frac{d_{ij}^2 + d_{ip}^2 - d_{jp}^2}{2 d_{ij} d_{ip}}\right)$$

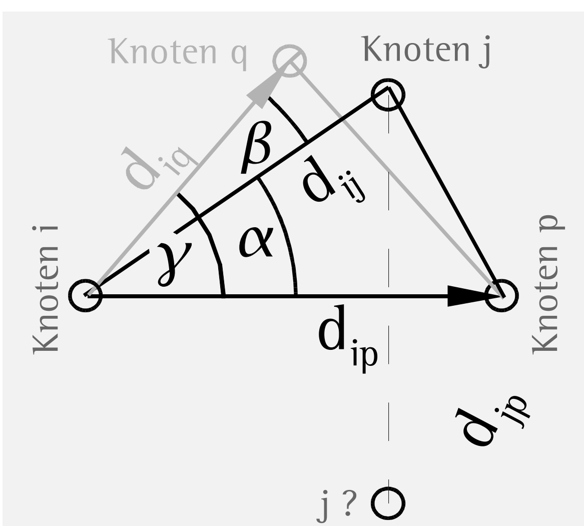


Localization: Bekannte Entfernungen ohne Positionen

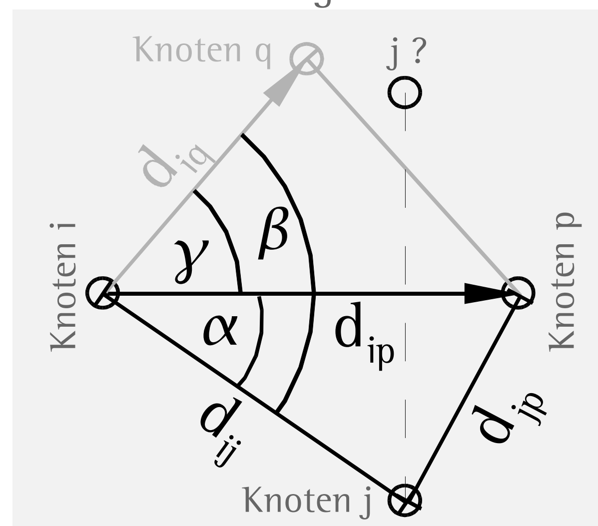
Bestimmung eines lokalen Koordinatensystems (3): Einbetten der sichtbaren Knoten

Ob der Knoten j über oder unter der durch i und p definierten X-Achse des lokalen Koordinatensystems liegt kann dadurch abgeschätzt werden, wie sich beta zusammen setzt:

Fall: beta = gamma-alfa



Fall: beta = alfa+gamma



$$\beta \approx |\alpha + \gamma| \Rightarrow j_y = -d_{ij} \sin(\alpha)$$
$$\beta \approx |\gamma - \alpha| \Rightarrow j_y = +d_{ij} \sin(\alpha)$$

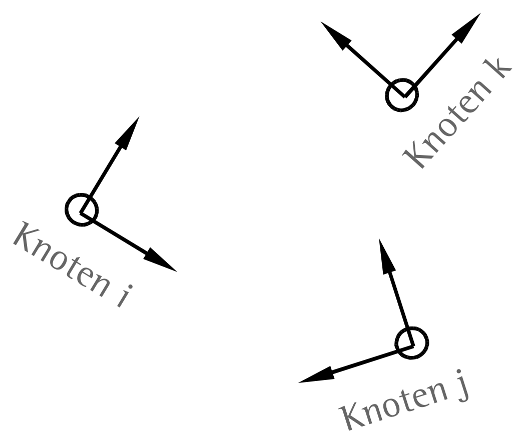
Localization: Bekannte Entfernungen ohne Positionen

Konsolidierung der Koordinatensysteme (4)

Ein Knoten k sitzt immer im Ursprung (0,0) seines eigenen, lokalen Koordinatensystems. Alle Knoten innerhalb seines Local View Sets (LVS_k) werden in Bezug zur Position von k interpretiert. Wo sich k jedoch innerhalb der Weltkoordinaten befindet ist unbekannt (und bleibt es auch).

Jedoch wird k auch von den anderen Knoten im Rahmen deren Koordinatensystemen interpretiert. In der unteren Abbildung liegt k auch im LVS von i, hat bezogen auf das KS von i also eine feste Position. Auch in Bezug auf das Koordinatensystem von j hat k eine Position – wenn auch eine andere als in Bezug zu i.

Nun muss noch ein Weg gefunden werden die lokalen Koordinatensysteme zum globalen des Netzwerks zu vereinigen. Hier sollen alle Knoten begl. des Knotens i interpretiert werden. Dazu müssen die KS gleichgerichtet werden. Als gleichgerichtet gelten sie dann, wenn ihre Achsen in die gleiche Richtung zeigen.

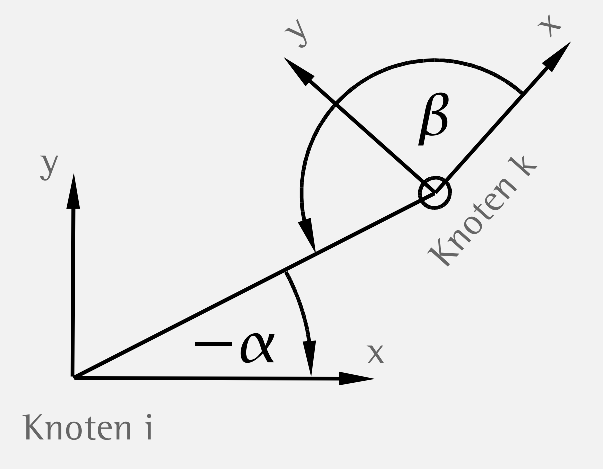


Fall 1: Die Gleichrichtung beinhaltet eine Drehung
Fall 2: Die Gleichrichtung beinhaltet eine Spiegelung
(meist zusätzlich eine Drehung)

Localization: Bekannte Entfernungen ohne Positionen

Konsolidierung der Koordinatensysteme (4): 1. Fall (Angleichen d. Ausrichtung)

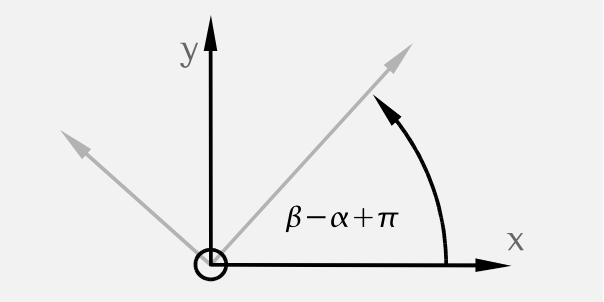
Alle Knoten im LVS von Knoten i sind ohnehin bereits im KS des Netzwerkes (mit Zentrum i) enthalten. Interessant ist nun auch die mittelbaren 2 Hop Knoten zu erfassen, die von i aus nicht unittelbar gesehen werden können (jedoch z. B. von k). Hierzu müssen die anderen lokalen KS so korrigiert werden, dass auch deren Kinder in das globale einbezogen werden können.



Die Achsen des Knoten k werden bereits als Sinus- bzw. Cosinus eines Winkels angegeben. Dieser Winkel muss nun korrigiert werden um:

$$\beta - \alpha + \pi$$
$$(\pi = 180°)$$

Drehung, die alle Knoten des LVS von k vollziehen müssen



Nach der Drehung müssen die Kinder des LVS von k noch um die Position von k selbst verschoben werden, um ihre Positionen in Bezug zu i zu finden.

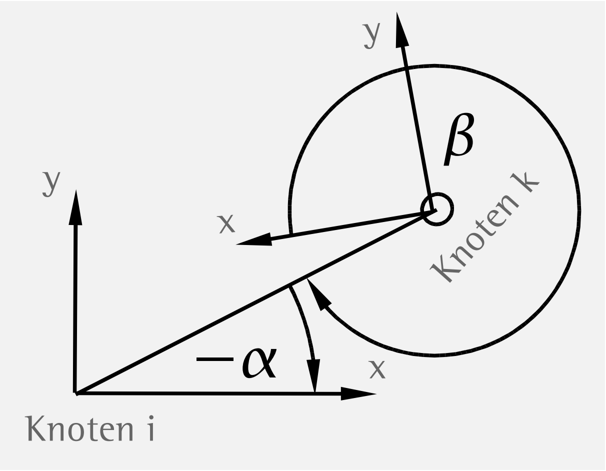
Bemerkung: Gedreht wird immer mathematisch, also gegen den Urzeigersinn.

Localization: Bekannte Entfernungen ohne Positionen

Konsolidierung der Koordinatensysteme (4): 2. Fall (Spiegelung einer Achse)

Nicht immer reicht eine einzelne Drehung, um die Koordinatensysteme gleichzurichten. Dies gilt immer dann, wenn nur die X- bzw. nur die Y-Achse des lokalen KS in die entgegengesetzte Richtung zeigt wie die entsprechende Achse des globalen KS. Auch bzgl. der Korrekturdrehung ergibt sich ein leicht veränderter Fall:

$$\beta - \alpha$$



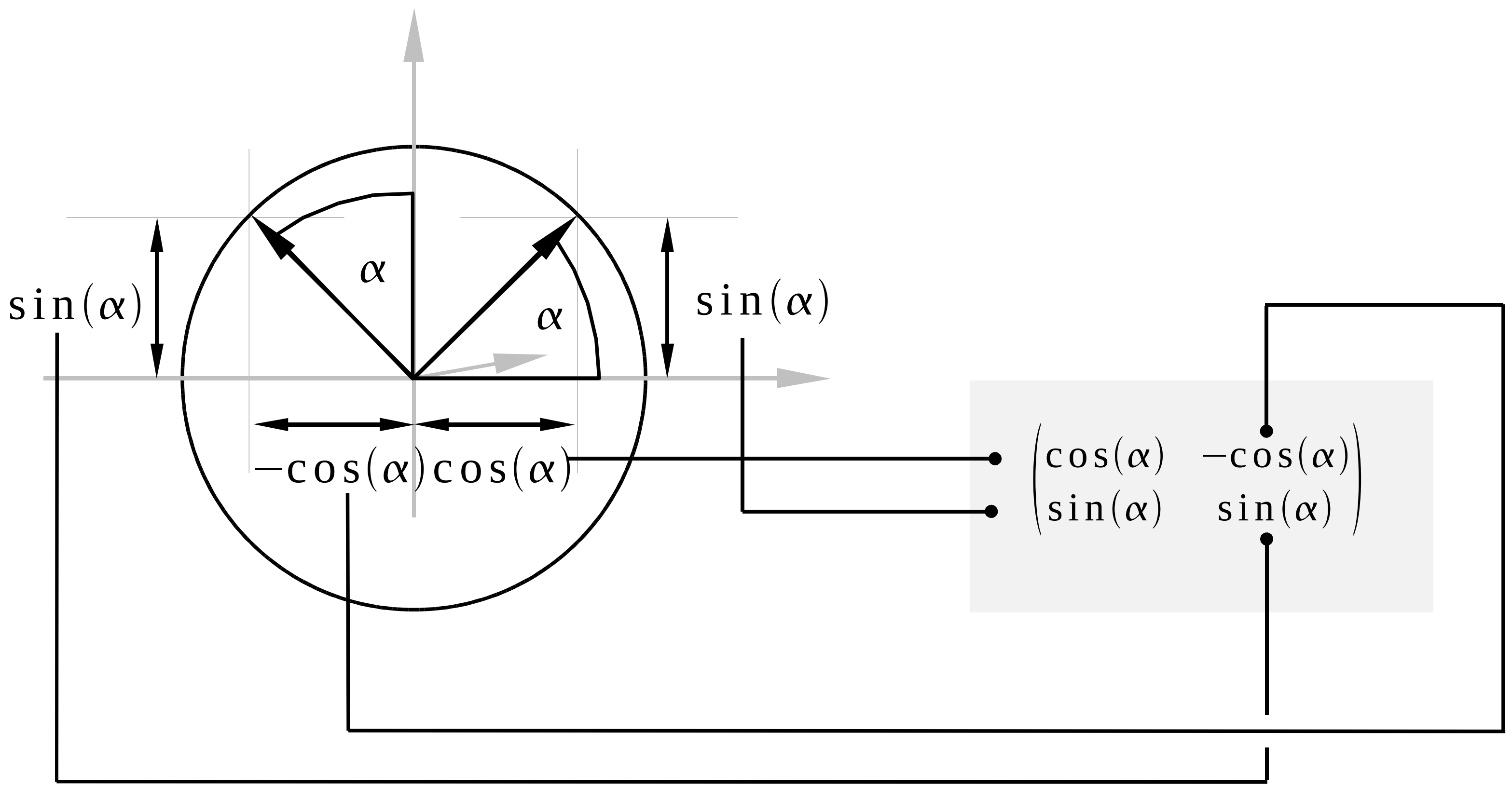
Danach folgt eine Spiegelung der X-Achse am Ursprung (d. h. alle X-Komponenten der Koordinaten von Knoten im LVS von k werden mit (-1) multipliziert.

Wie im ersten Fall auf der vorherigen Seite wird zur so berechnete Koordinate noch die Position des Knoten k innerhalb des globalen Koordinatensystems hinzuaddiert um die globale Koordinate eines Kindes von k zu erhalten.

Zur Erinnerung: Im LVS (Lokal View Set von k befinden sich alle seine Kinder).

Auffrischung: Projektionen und Koordinatensysteme

Sinus- Cosinus- Drehungs- Erinnerungsfolie



Localization: Verbesserte Lokalisierung

Fall: Es gibt mehrere Knoten, deren globale Position bekannt ist

Unbekannt ist die Position p_n des Knoten n. Im LVS von n befinden sich mehrere (hier 5) Knoten mit bekannter Position. Man möchte nun a) alle Knoten in die Positionsberechnung für n einbeziehen und b) die Positionen der Knoten stärker gewichten, zur denen n seine Entfernung besonders gut abschätzen kann. Da die Feldstärke eines empfangenen Signals zum Quadrat der Entfernung abnimmt, kann die Entfernung der nächstliegenden Knoten besonders exakt abgeschätzt werden.

Gegeben

n: Neuer Knoten mit unbekannter Position

1-5: Knoten mit bekannter Position (allgemein Index k)

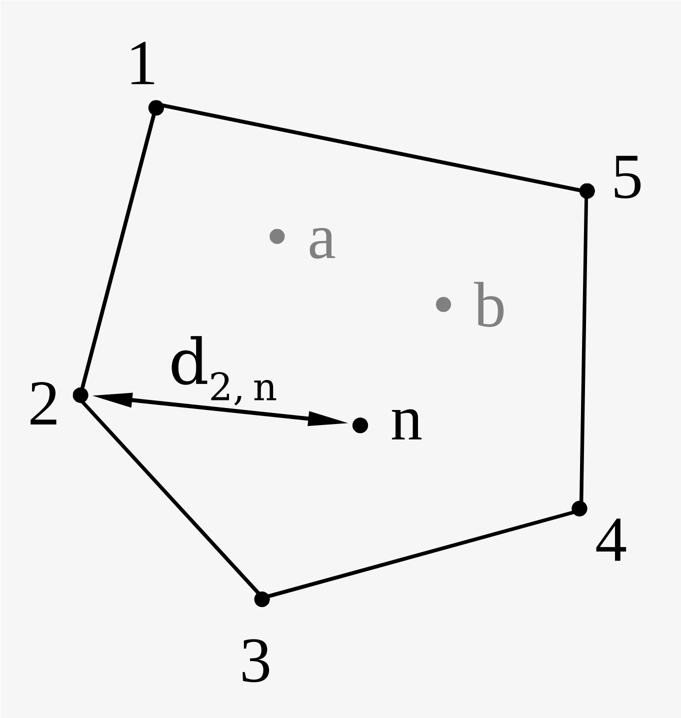
a, b: Weitere Knoten, nicht auf der konvexen Hülle. Sie werden nicht berücksichtigt.

p_k: (Globale) Position von Knoten k

$d_{k,n}$: Entfernung zwischen Knoten k und n. Wird aus der Feldstärke des eintreffenden Signales geschätzt.

Gesucht

p_n: Position von Knoten n.



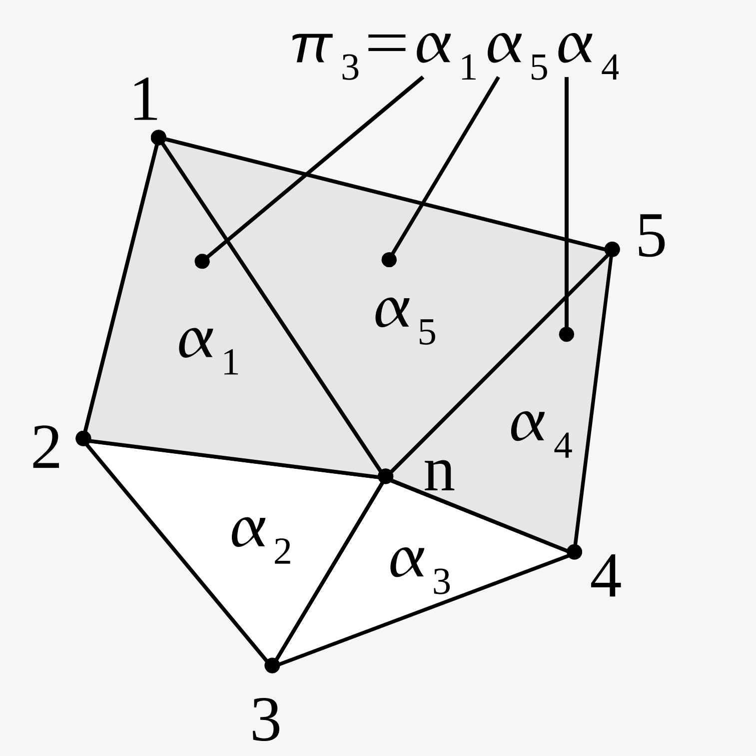
Localization: Verbesserte Lokalisierung

Fall: Es gibt Knoten, deren globale Position bekannt ist

Die Position p_n soll als Konvexkombination der benachbarten und bereits lokalisierten Knoten $p_1,\dots, p_5$ berechnet werden. Frage: Wie sind die Gewichte l_k zu wählen, damit eine möglichst optimale „Mischung" der bekannten Positionen der wahren Position von n möglichst nahe kommt? Idee: konvergiert der Knoten n gegen den Knoten 3 in der unteren Abbildung, so soll auch die durch Knoten 3 abgeschätzte Position von n besonders stark gewichtet werden.

$$p_n = \sum_{k=1}^{K} l_k \, p_k$$

$$\sum_{k=1}^{K} l_k = 1$$



Localization: Verbesserte Lokalisierung

Motivation u. Klassifikation
Keine Positionen keine Entfern.
Einfache globale Positionierung
Einige Positionen keine Entf.
Keine Positionen bekannte Entf.
Verbesserte globale Position.

Fall: Es gibt mehrere Knoten, deren globale Position bekannt ist

Grundlage für die Berechnung der Gewichte l_k sind die Flächeninhalte der Dreiecke, die der noch unlokalisierte Knoten n und jeweils zwei benachbarte lokalisierte Knoten aufspannen.

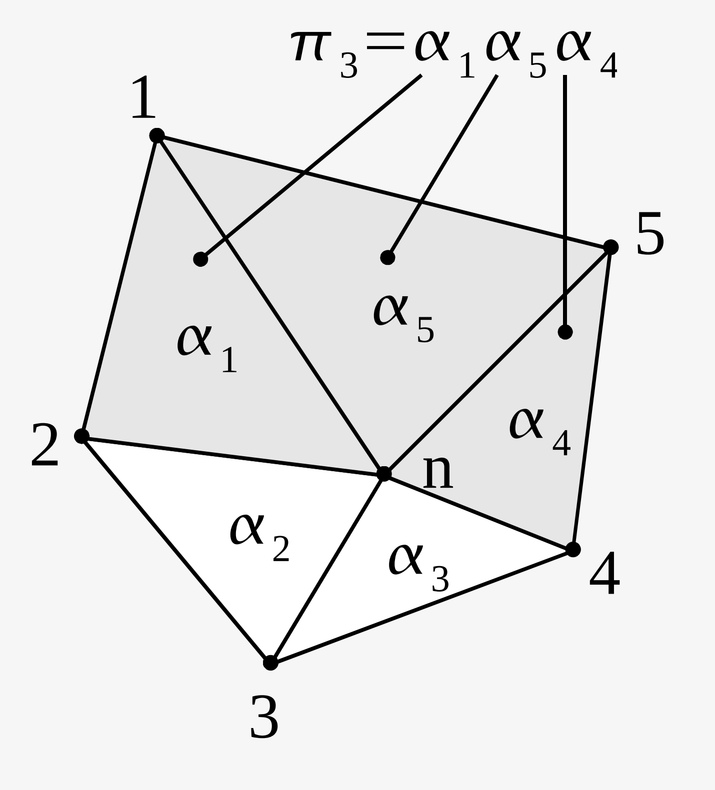
$$\alpha_k = \sqrt{(d_{k,k+1} + d_{k,n} + d_{k+1,n})(d_{k,n} + d_{k+1,n} - d_{k,k+1})(d_{k+1,n} + d_{k,k+1} - d_{k,n})(d_{k,k+1} + d_{k,n} - d_{k+1,n})}$$

Die Flächeninhalte alpha_k können allein anhand der Seitenlängen der Dreiecke berechnet werden.

Das eigentliche Gewicht einer Knotenposition k setzt sich dann aus dem Produkt fast aller Dreiecksflächen zusammen.

$$\pi_k = \frac{\prod_{i=1}^{K} \alpha_i}{\alpha_{k-1}\alpha_k}$$

Ausgenommen sind lediglich die beiden Dreiecke, die an die Gerade zwischen Knoten n und k (rechts Knoten 3) angrenzen.



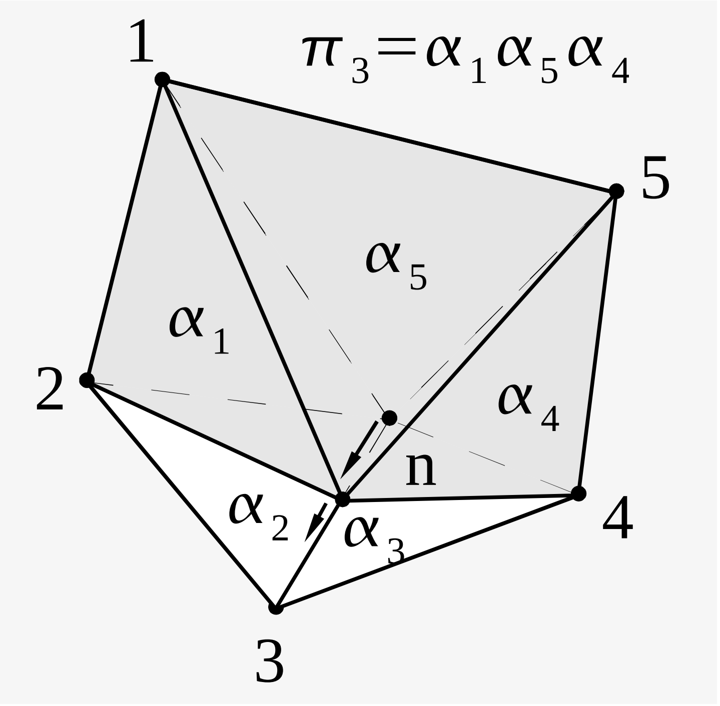
Localization: Verbesserte Lokalisierung

Fall: Es gibt Knoten, deren globale Position bekannt ist

Warum gehen gerade die beiden Flächen alfa_2 und alfa_3 nicht in das Gewicht des Knoten 3 ein, zumal allein diese Flächen Knoten 3 berühren?

Antwort: Konvergiert n im Beispiel unten gegen Knoten 3, so konvergiert das Gewicht PI_3 gegen das Gewicht des gesamten Polygons. Folglich geht die Position p_3 immer stärker in die Berechnung der Position p_n ein. Die Gewichte der anderen Knoten konvergieren dagegen immer stärker gegen Null. Dies ist insofern gewollt, als bei Annäherung von Knoten n an Knoten 3 auch dessen Position dominieren soll und gleichzeitig die der anderen aus der Gewichtung (zunehmend) herausfallen müssen.

Die PI_n eignen sich jedoch nicht unmittelbar als Gewichte. Zuvor muss noch eine Normierung erfolgen, die sicherstellt, dass die Summe der Gewichte tatsächlich 1 ergibt (ansonsten würde p_n nicht innerhalb des Polygones liegen.



Localization: Verbesserte Lokalisierung

Fall: Es gibt Knoten, deren globale Position bekannt ist

Die Normierung der Gewichte PI zu den Gewichten l der Konvexkombination wird wie folgt bewerkstelligt:

$$l_k = \frac{\pi_k}{\sum_{i=1}^{K} \pi_i}$$

Dabei ändert sich am Verhältnis der Gewichte zueinander nichts. Nur die Summe der l ergibt 1, die Summe der PI kann gegeben beliebig sein. Nun hätte man gern, dass die durch l gewichteten Positionen p_k auch gleich die bisher unbekannte Position des inneren Knoten n ergibt:

$$\cancel{p_n = \sum_{k=1}^{K} l_k \, p_k}$$

Aus didaktischen Gründen wurde bisher unterschlagen, dass dies nicht der Fall ist. (Siehe hierzu auch das Beispielprogramm).

Statt also die Positionen der Knoten p_k einfach aufzusummieren, muss jeder dieser Knoten die wahre Position von n abschätzen. Die finale Position besteht dann nicht aus der gewichteten Summe der Knotenpositionen selbst (wie in der Gleichung oben), sondern aus den Abschätzungen der Knoten.

Localization: Verbesserte Lokalisierung

Fall: Es gibt Knoten, deren globale Position bekannt ist

Wie bereits am Anfang des Kapitels gezeigt reichen zwei Knoten aus, um zwei Kandidaten für die Position eines dritten Knoten zu berechnen. Da hier nur die Kandidaten im Inneren des Polygons interessieren, kann der Kandidat außerhalb leicht ausgeschlossen werden.

Hier soll nun Knoten k die Position von Knoten n abschätzen. Dazu braucht er einen weiteren Knoten. Er könnte sich daher für seinen Nachbarn k-1 oder k+1 entscheiden. Noch besser und wie unten gezeigt ist es, beide Nachbarn in die Positionsberechnung einzubeziehen. Dabei schlägt k einen Kreis um sich selbst, der n tangiert, also mit dem Radius $d_{n,k}$ (~ über d in der Skizze unten, da es sich nur um eine Schätzung handelt). Dies tun auch Knoten k-1 und k+1. Wären alle Entfernungsschätzungen d exakt, so würde nur ein Schnittpunkt an der Position von n entstehen. Genaue Entfernungen sind aber kaum zu erwarten, so dass wie unten gezeigt zwei Schnittpunkte s_1 und s_2 entstehen. Die Mitte dieser beiden Schnittpunkte wird als Vorschlag von Knoten k für die Position von n angesehen und mit dem Gewicht l_k gewichtet. Die Gleiche Berechnung führen alle anderen Knoten mit ihren Nachbarn aus und gewichten ihr Ergebnis mit ihrem l, so dass am Ende ein optimiertes Ergebnis für die Position von n entsteht.