Effiziente Synthese konsistenter Graphen und ihre Anwendung in der Lokalisierung akustischer Quellen

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart
zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Abhandlung

vorgelegt von
Martin Kreißig
aus Stuttgart

Hauptberichter:	Prof. Dr.-Ing. Bin Yang
Mitberichter:	Prof. Dr.-Ing. habil. Rudolf Rabenstein
Tag der mündlichen Prüfung:	16.06.2015

Institut für Signalverarbeitung und Systemtheorie
der Universität Stuttgart

2015

Danksagung

Am liebsten würde ich sagen ‘Es ist doch nix dabei eine Dissertation zu schreiben. Jeder kocht nur mit Wasser und mit dem nötigen Fleiß – von dem ich vermutlich ausreichend hatte – schafft es jeder.’Allerdings ist man es ja nicht allein, der die Wissenschaft voranbringt, sondern der Austausch, die Diskussion und die Kritik von anderen...ja ja, die Kritik. Davon hatte mein Doktorvater Prof. Yang eine Menge. Aber nur durch seine konstruktiven Vorschläge habe ich es zu dem vorliegenden Ergebnis geschafft und dafür bin ich ihm sehr dankbar: für meine wissenschaftliche Weiterentwicklung, die Freiheiten die er mir gegeben hat und die ständige Unterstützung.

Der Austausch wurde auch in der Kooperation mit Prof. Rudolf Rabenstein und seinem wissenschaftlichen Mitarbeiter Paolo Annibale vom Lehrstuhl für Multimediakommunikation und Signalverarbeitung an der Friedrich-Alexander Universität Erlangen-Nürnberg gelebt. Vielen Dank für die erfolgreiche Zusammenarbeit und lehrreiche gemeinsame Zeit. Zusätzlich möchte ich Prof. Rabenstein herzlich danken für die Übernahme der Aufgaben des Mitberichters.

Zudem waren es die Kollegen Fabian Schmieder, Kilian Rambach, Roman Streubel, Thomas Küstner und Christof Zeile, die meine Erstfassung überlebt haben: Danke für das Korrekturlesen und die Kommentare. Das Gleiche gilt für Frank Seeger, Fabian Fertig und David Rörich, die neben ihrer Arbeit noch die Zeit gefunden haben ihren Input zu liefern. Außerdem möchte ich euch allen, Patrick Häcker, Stefan Uhlich, Benedikt Lösch, Christian Würslin und Fabian Friedrichs für die ständige Diskussion und Unterstützung in vielerlei Hinsicht danken. Letzterem und Danny Krautz will ich vor allem für die Gespräche vor meiner Dissertation danken, sonst hätte ich wahrscheinlich nie angefangen.

Aber was wäre die Arbeit am Institut für Signalverarbeitung und Systemtheorie für seine Assistenten ohne die tatkräftigen Helfer im Hintergrund? Danke Elisabeth, René und Michael für die Unterstützung bei so vielen alltäglichen Problemchen.

Wie habe ich es überhaupt bis zur Promotion geschafft? Ohne meine Eltern hätte ich es jedenfalls nicht geschafft. Danke für euer Vertrauen, eure Offenheit und Unterstützung, sonst wären so einige Punkte in meinem Werdegang sicher anders verlaufen.

Zuletzt gilt mein Dank meiner wunderbaren Frau. Ohne ihre Nachsicht, Unterstützung und Fürsorge hätte ich vor allem die letzten Wochen nicht so unbeschadet überstanden. Danke mein Schatz.

Inhaltsverzeichnis

Abkürzungsverzeichnis

BFS	Breitensuche, engl. Breadth-First-Search.
BSS	Blinde Quellentrennung, engl. Blind Source Separation.
BT	Rückverfolgungsverfahren, engl. Backtracking.
CC-Graph	Kompatibilitäts-Konflikt-Graph, engl. Compatibility-Conflict-Graph.
CLS	Beschränkte kleinste Quadrate, engl. Constrained Least Squares.
CRS	Cramer-Rao-Schranke.
DATEMM	Disambiguation of TDOA Estimation in Multipath, Multisource environments.
DFS	Tiefensuche, engl. Depth-First-Search.
DFT	Diskrete Fourier-Transformation.
FM	Fundamentale Maschen.
GCC	Verallgemeinerte Kreuzkorrelation, engl. Generalized Cross-Correlation.
GN	Gauß-Newton Verfahren.
ICA	Independent Component Analysis.
ISM	Spiegelquellenmethode, engl. Image-Source-Model.
LS	Kleinste Quadrate, engl. Least Squares.
ML	Maximum Likelihood.
MSE	mittlerer, quadratischer Fehler, engl. Mean Squared Error.
PHAT	Phasentransformationsfilter.
SCT	State Coherence Transform.
SNR	Signal-zu-Rauschabstand, engl. Signal-To-Noise Ratio.
SONG	Synthese konsistenter Graphen, engl. synthesis of consistent graphs.
SRP	Steered Response Power.
TDOA	Laufzeitdifferenz, engl. Time Difference of Arrival.
TOA	Laufzeit, engl. Time of Arrival.

ULS	Unbeschränkte kleinste Quadrate, engl. Unconstrained Least Squares.
WDF	Wahrscheinlichkeitsdichtefunktion.

Symbolverzeichnis

$\mathbf{A}$	Adjazenzmatrix eines Graphen.
B	Länge eines Signalblocks.
c_0	Schallgeschwindigkeit.
d_{ij}	Exakte Abstandsdifferenz zwischen den Mikrofonen i und j.
δ_{ij}	Zufallsvariable der gemessenen Abstandsdifferenz zwischen den Mikrofonen i und j.
D	Dimension des kartesischen Koordinatensystems.
$\bar{E}$	Menge der Konfliktkanten im CC-Graph.
e_c	Schwellwert für den Schallgeschwindigkeitsfehler.
E	(kompatible) Kantenmenge.
ϵ	Schwellwert für die ungefähre Konsistenz.
E	Erwartungswert einer Wahrscheinlichkeitsdichtefunktion.
e_t	Schwellwert für den residualen TDOA Fehler.
f_A	Abtastfrequenz.
G_{cc}	Kompatibilitäts-Konflikt-Graph (CC-Graph).
G_{KB}	Komplementärer Baum eines Graphen.
G	Gerichteter und ungewichteter Graph.
G_{AB}	Aufspannender Baum eines Graphen.
G^w	Gerichteter und gewichteter Graph.
G^{w_k}	Graph mit konsistenter Kantengewichtsbelegung $\underline{w}_k$.
$\mathbf{I}$	Einheitsmatrix.
$\hat{K}$	Anzahl der synthetisierten, konsistenten Graphen.
K_n	Anzahl der TDOA-Maxima in der GCC der Mikrofone $n = (i, j)$ bzw. Kardinalität der Kantengewichtsmenge der Kante e_n.
$\tilde{K}_i$	Anzahl konsistenter Gewichtskombination der i-ten Masche.
$\mathbf{L}$	Maschenmatrix.
$\mathbf{L}_{\mathrm{FM}}$	Fundamentale Maschenmatrix.
$\underline{l}$	Topologische Masche der Form $\{-1, 0, 1\}^N$.
$\underline{l}_{f,i}$	Topologische i-te fundamentale Masche.

$\underline{l}_{f,i}^{w_k}$	k-te konsistente Kantengewichtskombination der i-ten fundamentalen Masche.
M	Anzahl der Mikrofone bzw. Knoten.
$\underline{m}_i$	Position des i-ten Mikrofons.
N_{FM}	Anzahl fundamentaler bzw. unabhängiger Maschen.
N	Anzahl der Mikrofonpaare bzw. Anzahl der Kanten.
N_i	Anzahl der Kanten in der i-ten Masche.
N_{L}	Anzahl der Maschen in einem vollständig zusammenhängenden Graphen.
N_θ	Anzahl gültiger Kanten in einem partiell konsistenten Graphen.
P_D	Detektionsrate der gesuchten konsistenten Graphen.
$p_{\bar{E}}$	Relative Anzahl in Konflikt stehender Kanten im CC-Graph.
p_E	Relative Anzahl kompatibler Kanten im CC-Graph.
P_F	Falschalarmrate der durchschnittliche zusätzlich gefundenen konsistenten Graphen.
P_{iq}	Anzahl der Reflexionen von der Quelle q zum i-ten Mikrofon.
$\underline{p}$	Position einer Quelle.
Q	Anzahl der Quellen.
ρ	Zufallsvariable des Rauschen.
$\hat{\rho}$	Fehler in der TDOA-Schätzung.
$\mathbb{R}$	Menge der reellen Zahlen.
S	Anzahl der Simulationsdurchläufe.
T_{60}	Nachhallzeit.
T_A	Abtastintervall.
$\hat{\tau}_{ij}$	geschätzter TDOA-Wert am Mikrofonpaar (i, j).
$\tau_{ij,q,\mu\nu}$	Zufallsvariable der geschätzten Laufzeitdifferenz zwischen den Mikrofonen i und j des Quellsignals der q-ten Quelle über die Ausbreitungspfade μ und ν.
$t_{iq,\mu}$	Laufzeit des Signals von Quelle q zum Mikrofon i über den Pfad μ.
$t_{ij,\mu\nu}$	Exakter TDOA-Wert am Mikrofonpaar (i, j) über die Ausbreitungspfade μ und ν.
V_{cc}	Knotenmenge des CC-Graphen.

V	Knotenmenge.
$\underline{w}$	Gültige Kantengewichtsbelegung.
W_n	Kantengewichtsmenge zur Kante e_n.
$x_i(t)$	Signal am Mikrofon i.
$\mathbb{Z}$	Menge der ganzen Zahlen.
$\mathbf{Z}$	Inzidenzmatrix eines Graphen.

Notation

$\{\cdot\}$	Eine ungeordnete Menge
$(\cdot)$	Eine geordnete Menge
$[\cdot]$	Geordnete Menge über einem Körper (Vektor)
$[\cdot]^T$	Transponierter Vektor oder Matrix
$[\cdot]^+$	Pseudoinverse einer Matrix
$(\cdot)^*$	Komplexe Konjugation
$\underline{x}$	Spaltenvektor
$\mathbf{X}$	Matrix
j	Imaginäre Einheit

Abstract

Many signal processing and sensor fusion applications are based on the estimation of difference measurements between pairs of nodes or sensors such as time difference of arrival (TDOA) for localization, voltage as the difference of electrical potentials and relative velocity and position measurements in automotive assistance systems. In practice, these measurements are erroneous, ambiguous and noisy and lead to mismatches in-between different sensor pairs. This thesis focuses on the combinatorial problem of assigning the correct difference measurements as weights in the graph of sensor nodes. All problems share the mathematical property, that the sum of difference measurements along a loop of nodes is zero, which is called the zero cyclic sum condition. A graph of matching difference measurements fulfills this condition along all possible loops. Such a graph is called a consistent graph.

As one out of many possible applications of the synthesis of consistent graphs (SONG) algorithm, this thesis revises the engineering problem of locating many acoustic sources simultaneously in a reverberant environment. In chapter 2 we give an introduction to the different methods and principles of acoustic source localization. Especially, the TDOA-based localization, that has the ambiguity of assigning the correct TDOA to the corresponding source, is discussed in detail. Based on this example we further introduce the mathematical framework of graph theory in chapter 3 and provide the algorithms commonly known and used throughout this thesis. This includes the algorithm of Hopcroft and Tarjan [Hopcroft u. Tarjan, 1973] to find biconnected components which are subgraphs with at least one loop or the depth-first-search (DFS) and breadth-first-search (BFS) to find a spanning tree in a given graph. The spanning tree is necessary to determine the set of fundamental loops, that form the smallest set of loops to check the cyclic sum condition. In chapter 4, the synthesis of consistent graphs is performed by merging the consistent fundamental loops in a bottom-up way in order to find all possible consistent loop combinations or consistent graphs, respectively. Therein, two options turn out to be useful:

- The synthesis of fully consistent graphs which returns a set of consistent graphs that have a valid consistent edge weight on all edges.
- The synthesis of partially consistent graphs that may lack some weights. This is necessary in applications with partially missing difference measurements.

Both options are based on determining the subsolutions of consistent loops in the sen-

sor graph. The first option is then solved by a backtracking (BT) algorithm. The second option is derived by introducing a new type of graph, the Compatibility-Conflict-graph (CC-graph), that is able to express three different types of relations between the subsolutions of consistent loops:

- Two consistent loops do not intersect with their edges.
- Two consistent loops do intersect with common edge weights (compatible).
- Two consistent loops do intersect with different edge weights (conflict).

In order to synthesize all possible partially consistent graphs, a new algorithm, namely CCGsearch, is introduced and is proven to work correctly. The computational complexities of both synthesis approaches – BT and CCGsearch – are derived mathematically and verified by simulations.

Due to malicious initialization of the SONG algorithm, a degradation in the efficiency and detection of consistent solutions may occur. These shortcomings are mainly related to the topology of the sensor graph and can be solved by different workarounds, proposed and validated regarding their recognition capabilities and efficiency in chapter 5. It turns out, that a spanning tree with additional information about the reliability of the measured signals performs best. This is achieved by the spanning tree algorithm of Kruskal and Prim [Kruskal, 1956], which requires an additional quality value to all edges. Thus, several possibilities are shown to adapt the SONG algorithm to the application in order to obtain the best results.

Finally, in chapter 6, simulations and measurements of the SONG algorithm are performed and the adaptation to the acoustic source localization problem is done. Therefore, we introduce the approximate consistency condition that allows to process noisy edge weights. The results of the localization are compared with the steered response power (SRP) localization approach with a phase transform, that applies an additional whitening filter. It turns out that SONG performs equal in terms of the detection rate, while consuming less computation time. Moreover the TDOA-based localization techniques have more accurate position estimates than the grid search of SRP-PHAT.

Kurzfassung

In vielen Ingenieursanwendungen werden relative Differenzmessungen zwischen Sensorpaaren verwendet, wie z.B. die Laufzeitdifferenzen zwischen Mikrofonen für die Lokalisierung von akustischen Quellen, elektrische Spannungen als Differenz elektrischer Potentiale und die Relativgeschwindigkeit und -abstände in Fahrerassistenzsystemen. Diese Differenzmessungen sind oft fehlerbehaftet, mehrdeutig und verrauscht und führen somit zu Fehlkombinationen zwischen unterschiedlichen Sensorpaaren. Die vorliegende Arbeit beschäftigt sich mit dem kombinatorischen Problem der korrekten Zuordnung dieser Differenzmessungen in einem Graph von Sensoren. Dabei entsprechen die Sensoren den Knoten des Graphen und die Differenzmessungen den Kantengewichten zwischen diesen Knoten. Alle genannten Probleme haben die gemeinsame mathematische Eigenschaft, dass die Summe von Differenzmessungen entlang einer Masche von Sensoren Null ergibt. Diese Eigenschaft ist als Nullsummenbedingung bekannt. Ein Graph aus Differenzmessungen vom gleichen Ursprung erfüllt diese Eigenschaft entlang jeder Masche und wird konsistent genannt.

In dieser Arbeit wird das Problem der simultanen, akustischen Mehrquellenlokalisierung in echobehafteten Umgebungen genauer betrachtet und daran beispielhaft die Anwendungsmöglichkeit der Synthese konsistenter Graphen (synthesis of consistent graphs, SONG) gezeigt und analysiert. Dafür werden in Kap. 2 zunächst die Grundlagen der akustischen Lokalisierung eingeführt und unterschiedliche Ansätze vorgestellt. Im Besonderen werden die laufzeitdifferenzbasierten Lokalisierungsverfahren betrachtet, die das Problem der uneindeutigen Zuweisung der Laufzeitdifferenzen zu den Quellen haben. Anhand dieses konkreten Anwendungsbeispiels der Lokalisierung wird die Problematik auf ein graphentheoretisches Problem abstrahiert. Deshalb werden in Kap. 3 die graphentheoretischen Grundlagen gelegt und bereits bekannte Algorithmen, wie sie in dieser Arbeit angewendet werden, eingeführt. Das umfasst den Algorithmus von Hopcroft und Tarjan, der zweifach-zusammenhängende Teilgraphen findet, die mindestens eine Masche beinhalten und die Tiefen- und Breitensuche, um einen aufspannenden Baum zu finden. Der aufspannende Baum ist notwendig, um die Menge der fundamentalen Maschen (FM) zu bestimmen, welche die kleinste Menge von Maschen zur Überprüfung der Nullsummenbedingung darstellt. Die Synthese konsistenter Graphen erfolgt in Kap. 4 durch das Zusammenführen der konsistenten FM, bis alle konsistenten Kantengewichtskombinationen gefunden sind. Dabei unterscheidet man folgende Vorgehensweisen:

- Die Synthese voll konsistenter Graphen, die jeder Kante des Eingangsgraphen ein konsistentes Kantengewicht zuweisen.
- Die Synthese partiell konsistenter Graphen, die nur eine Teilmenge von Kanten beinhalten. Dies ist notwendig, falls in einer Anwendung manche Differenzmessungen fehlerhaft sind.

Beide Vorgehensweisen basieren auf den konsistenten FM des Sensorgraphen, die die Nullsummenbedingung erfüllen. Der erste Ansatz wird über ein Rückverfolgungsverfahren (Backtracking, BT) realisiert. Das zweite Verfahren wird aus dem Kompatibilitäts-Konflikt-Graph abgeleitet, der ein neuer Typus von Graph ist und die drei unterschiedlichen Zustände zwischen den konsistenten FM beschreibt:

- Zwei konsistente Maschen haben keine gemeinsame Kanten.
- Zwei konsistente Maschen haben gemeinsame Kanten und identische Kantengewichte (kompatibel).
- Zwei konsistente Maschen haben gemeinsame Kanten und unterschiedliche Kantengewichte (Konflikt).

Um alle möglichen, partiell konsistenten Graphen zu synthetisieren, wird der neue Algorithmus CCGsearch eingeführt und auf Vollständigkeit bewiesen. Die Berechnungskomplexitäten der beiden Ansätze – BT und CCGsearch – werden sowohl analytisch hergeleitet als auch durch Simulationen verifiziert. Aufgrund einer schlechten Initialisierung des SONG-Algorithmus kann es allerdings zu einer längeren Rechenzeit und zum Verlust von Lösungen kommen. Das liegt hauptsächlich an der Topologie des Sensorgraphen und kann auf unterschiedliche Arten gelöst werden, wie sie in Kap. 5 diskutiert werden. Dabei werden vor allem die Detektionsrate und die Effizienz betrachtet. Es zeigt sich, dass besonders ein aufspannender Baum mit zusätzlichen Kenntnissen über die Qualität der Differenzmessungen das beste Ergebnis erzielt. Ein möglicher Algorithmus für einen solchen aufspannenden Baum ist von Kruskal und Prim gegeben [Kruskal, 1956]. Es werden somit viele Möglichkeiten gezeigt, den SONG-Algorithmus flexibel auf die zugrundeliegende Anwendung anzupassen.

In Kap. 6 zeigen Simulationen und Messungen die gute Effizienz von SONG und die Einbindung in den Anwendungsfall der akustischen Quellenlokalisierung. Dafür führen wir noch die näherungsweise Konsistenz ein, um verrauschte Differenzmessungen verarbeiten zu können. Diese Ergebnisse der Lokalisierung werden mit dem Verfahren der Steered-Response-Power (SRP) für die Mehrquellenlokalisierung mit Phasentransformationsfilter (PHAT) verglichen. Es wird gezeigt, dass SONG eine ebenso gute Detektionsrate wie SRP-PHAT erreicht bei einem geringeren Rechenaufwand. Zudem können die laufzeitdifferenzbasierten Lokalisierungsverfahren eine genauere Positionsschätzung erreichen, da sie nicht auf die räumliche Auflösung des Suchrasters angewiesen sind.

Kapitel 1

Einleitung

Es existieren einige Anwendungsbeispiele, in denen die Graphentheorie ein hilfreiches Werkzeug für das Lösen von Problemen der Ingenieurwissenschaften ist. Für das Berechnen der kürzesten Strecke in einem Straßennetz werden z.B. Algorithmen wie der von Bellman-Ford, Dijkstra oder die Breitensuche (Breadth-First-Search, BFS) verwendet, die ursprünglich für das Problem des Handelsreisenden, der alle Knoten in einem Graphen besuchen will, entworfen wurden [Cormen u. a., 2009]. Die vorliegende Arbeit beschäftigt sich hingegen mit der Problemstellung, aus einem Satz von möglichen Differenzmessungen, wie sie bei der Lokalisierung akustischer Quellen als Laufzeitdifferenzen (Time Differences of Arrival, TDOA) oder bei der Analyse elektrischer Netzwerke als Potentialdifferenzen vorkommen, die richtigen Kombinationen auszuwählen, die zu derselben Quelle bzw. Netzwerk gehören. Dafür wird das Problem auf einen Graphen mit Sensoren als Knoten und Differenzmessungen als Kantengewichte abstrahiert.

Mit der Einführung der Maschengleichung hat Kirchhoff bereits 1845 [Kirchhoff, 1845] die Grundlage für das vorliegende Themengebiet geschaffen. Dabei zeigt er, dass die Summe der elektrischen Spannungen entlang jeder Masche in einem Netzwerk Null ergibt. Während Kirchhoff die Analyse eines elektrischen Netzwerks durchführt, wird in dieser Arbeit ein Verfahren gezeigt, das anhand von gegebenen elektrischen Spannungen ein gültiges elektrisches Netzwerk synthetisiert. Diese Eigenschaft ist auf die Definition der Spannung als Potentialdifferenz zurückzuführen und ist somit auch für andere Anwendungen anwendbar, die auf Differenzmessungen beruhen.

Über die elektrischen Spannungen hinaus gehören auch die Laufzeitdifferenzen in der Lokalisierung akustischer Quellen mittels einer Mikrofongruppe zu den Anwendungen, die auf Differenzmessungen aufbauen. Dabei erreicht das von einer Quelle emittierte akustische Signal nach unterschiedlichen Laufzeiten die Mikrofone i und j. Mithilfe der Mikrofonpositionen und der Mikrofonsignale kann anschließend eine Schätzung der Quellposition erfolgen. Diese Positionsschätzung findet Anwendung in Bereichen wie der Geräuschunterdrückung, indem auf die Einfallsrichtung der Quelle fokussiert wird und andere Einflüsse unterdrückt werden [Dibiase u. a., 2001]. Durch Kenntnis der Quellposition kann zudem die Raumimpulsantwort und damit Informationen über die

Raumgeometrie gewonnen werden, was wiederum für eine weitere Qualitätsverbesserung des aufgenommenen Quellsignals genutzt werden kann. Die Kenntnis der Position unterstützt zudem Verfahren wie die blinde Signaltrennung [Loesch u. a., 2009], Spracherkennung [Huang u. Chi, 2012] oder Sprecheridentifikation [Chen u. a., 2007]. Die Position wird auch direkt für Anwendungen der automatischen Kameraführung [Huang u. a., 2000] oder Mensch-Maschine-Kommunikation genutzt [Boehme u. a., 1998]. Neben den Verfahren der Signalunterraumsuche mittels Independent Component Analysis (ICA) (Kap. 2.3.4) oder dem Beamforming (Kap. 2.3.5) stellen die TDOA-basierten Lokalisierungsverfahren eine rechenarme Alternative dar. Allerdings weisen sie das Problem der Mehrdeutigkeiten auf, das mit dem Disambiguation of TDOA Estimation in Multipath, Multisource environments (DATEMM) Verfahren (Kap. 2.3.3) teilweise gelöst wurde. In der vorliegenden Arbeit wird eine erweiterte, vollständige Lösung dieses Problems betrachtet.

Neben der akustischen Sprecherlokalisierung sind auch nautische und seismologische Lokalisierungsverfahren zu nennen [Liu u. a., 2012; Sidorovich u. Gershman, 1998], die auf Laufzeitdifferenzen beruhen und durch die Anwendung von Synthese konsistenter Graphen (synthesis of consistent graphs, SONG) zuverlässiger und robuster werden. Dabei wird eine bekannte akustische Sequenz gesendet, die nach Reflektion zu unterschiedlichen Laufzeiten an den Sensoren erfasst wird. Anschließend erhält man über die Laufzeitdifferenzen zwischen den Sensoren einen Rückschluss auf die eigene, relative Position zur Umgebung bzw. durch die unterschiedlichen Ausbreitungspfade einen Rückschluss auf die Beschaffenheit des Untergrunds. Aufgrund des bekannten Aussendezeitpunktes bei diesen Verfahren mittels Reflektion kann zusätzlich die direkte Laufzeit verwendet werden, um die Ergebnisse zu verbessern.

Ein weiterer Einsatzbereich von SONG ist die verlässliche Schätzung von Positions- und Geschwindigkeitsdifferenzen in Fahrerassistenzsystemen [Carevic, 2007]. Der relative Abstand zwischen unterschiedlichen Fahrzeugen wird aus den Distanzinformationen des Radar- oder Videosystems bestimmt [Yang u. Kreißig, 2011]. Kombiniert man diese relativen Abstands- und Geschwindigkeitsinformationen zwischen mehreren Fahrzeugen, erhält man einen verlässlichen Gesamtüberblick der Verkehrssituation. Daraus können präventiv Gefahrsituationen erkannt und auch Navigationsmaßnahmen eingeleitet werden. Hierbei müssen die gleichen Relativwerte zwischen den Fahrzeugen kommuniziert und richtig kombiniert werden, wobei die Abstandsdifferenzen von mehreren Fahrzeugen sich zu Null addieren und somit SONG eingesetzt werden kann.

In Kap. 2 werden unterschiedliche Ansätze der laufzeitdifferenzbasierten Lokalisierungsverfahren beschrieben und auf die Problemstellung der TDOA Zuordnung zu den einzelnen Quelle in reflektionsbehafteten Mehrquellen-Szenarien eingegangen. Anschließend führen wir in Kap. 3 die notwendigen Grundlagen der Graphentheorie ein und damit das Konzept der konsistenten Graphen. In Kap. 4 wird der Algorithmus zur Synthese konsistenter Graphen eingeführt und dessen Vollständigkeit gezeigt. Es folgt eine

Diskussion der Komplexität des Algorithmus, die mittels Simulationen gestützt wird. Anschließend analysieren wir in Kap. 5 die Abhängigkeiten von der Topologie des Eingangsgraphen und damit die Robustheit des SONG-Algorithmus gegenüber schlechten Eingangsdaten. In Kap. 6 untersuchen wir zunächst mittels Simulationen und Messungen die Probleme in der TDOA-Schätzung bei der simultanen Mehrquellenlokalisierung. Danach wenden wir SONG an und analysieren die Ergebnisse bzgl. Erkennungs- und Verlustrate. Abschließend zeigt Kap. 7 eine Zusammenfassung und einen Ausblick der Anwendung des SONG-Algorithmus.

Kapitel 2

Methoden der Sprecherlokalisierung

Dieses Kapitel gibt einen Überblick über die möglichen Lokalisierungsstrategien. Zunächst werden die einfacheren Fälle mit einer Quelle betrachtet, wobei ein besonderer Fokus auf die TDOA-basierten Verfahren gelegt wird. Anschließend werden Probleme in reflektionsbehafteten Umgebungen und der simultanen Lokalisierung mehrerer Quellen betrachtet.

2.1 Lokalisierung einer Quelle

Um aus den Mikrofonsignalen einer Mikrofongruppe einen Rückschluss auf die Quellposition zu gewinnen, werden in der Literatur unterschiedliche Ansätze genannt. Hierbei werden räumliche Eigenschaften vorausgesetzt, wie eine konstante Schallgeschwindigkeit c_0 entlang des Ausbreitungspfades und eine direkte Sichtverbindung zwischen der Quelle und den Mikrofonen. Ein einfaches Lokalisieruungsverfahren ist der Beamformer, welcher nach einem Einfallswinkel mit maximaler Energie sucht [Dibiase u. a., 2001]. Er sucht dabei über ein Raumraster, dessen Genauigkeit den Aufwand entsprechend bestimmt. Besonders für breitbandige Signale ist der Rechenaufwand aufgrund des größeren spektralen Bereichs höher und somit nur bedingt für den Echtzeiteinsatz geeignet [Scheuing, 2007]. Alternative Ansätze sind die Signalunterraumanalyse wie Multiple Signal Classification (MUSIC) [Schmidt, 1986] oder Signaltrennverfahren wie die ICA [Loesch, 2013]. Die ICA wird dabei für die Suche nach den Signal- und Rauschkomponenten eingesetzt, wie sie für die Blinde Quellentrennung (blind source separation, BSS) benötigt werden. Die Positionsschätzung ist dabei ein wichtiger Bestandteil und kann als Beiprodukt verwendet werden. Diese Verfahren haben wie das Beamforming einen hohen Rechenaufwand und sind damit nicht zweckmäßig für eine echtzeitnahe Umsetzung.

Im Gegensatz dazu stellt die Lokalisierung über Laufzeiten (Times of Arrival, TOA) oder TDOA eine effizientere Lokalisierungsmethode dar, da mittels der gemessenen TOA bzw. TDOA die Quellpositionen direkt berechnet werden können. Deshalb werden diese Verfahren im Folgenden genauer betrachtet.

TOA und TDOA Verfahren In Abb. 2.1 ist der Aufbau eines Lokalisierungssystems gezeigt, wobei die Mikrofonpositionen mit $\underline{m}_i \in \mathbb{R}^D$ und die unbekannte Quellposition mit $\underline{p} \in \mathbb{R}^D$ gekennzeichnet sind. Die Dimension D kann 2 oder 3 sein. Die TOA t_i werden direkt zwischen der Quelle und dem jeweiligen Mikrofon ermittelt und die TDOA als Laufzeitdifferenz zwischen jeweils zwei Mikrofonen.

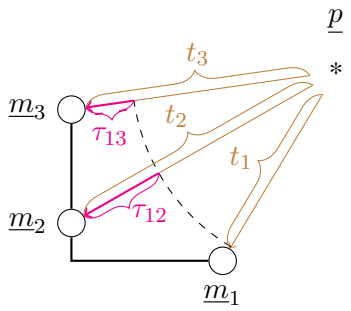


Abbildung 2.1: Aufbau eines TOA- und TDOA-basierten Lokalisierungssystems im Nahfeldmodell mit den Mikrofonpositionen $\underline{m}_1, \underline{m}_2, \underline{m}_3$ und der Quellposition $\underline{p}$. Die Lokalisierung kann sowohl über TOAs t_1, t_2, t_3 als auch über TDOAs τ_{12}, τ_{13} erfolgen.

Wie in [Shen u. a., 2008] und [Beck u. a., 2008] gezeigt, weisen die TOA-basierten Verfahren eine höhere Genauigkeit auf als die TDOA-basierten. Allerdings benötigen die TOA-Verfahren A-priori-Kenntnisse, um den Sendezeitpunkt zu bestimmen. Dies kann zum einen eine globale Uhr sein, auf die alle Sensoren synchronisiert werden oder Systeme wie das Global Positioning System (GPS), welche eine vordefinierte Sequenz senden, anhand derer die Laufzeiten z.B. vom Satelliten zum Empfänger ermittelt werden. Ein ähnliches Verfahren stellt das Sound Navigation and Ranging (SONAR) Verfahren dar, welches ein Tonsignal zu einem festen Zeitpunkt aussendet und nach der Reflexion am Objekt die doppelte Laufzeit empfängt. Die TOA-basierten Verfahren benötigen demzufolge zusätzlichen Hardwareaufwand und sind für natürliche Quellsignale wie Sprache, deren Sendezeitpunkt nicht bekannt ist, ungeeignet. Deshalb werden in dieser Arbeit die TDOA-basierten Lokalisierungsverfahren untersucht.

Nah- und Fernfeld-Modell Für die Lokalisierungsverfahren unterscheidet man zwischen dem Modell des Nah- oder Fernfeldes. Im Nahfeldmodell wie in Abb. 2.1 gezeigt, befindet sich die Quelle in Relation zum Durchmesser der Mikrofongruppe nahe dem Zentrum der Mikrofongruppe. Im Modell des Fernfeldes ist der Abstand der Quelle zur Mikrofongruppe um vieles größer als ihr Durchmesser. Unter diesen Bedingungen wird die einfallende, akustische Welle als eine Planarwelle modelliert. Wie in Abb. 2.2 skizziert, ist es somit nicht mehr möglich die Entfernung zur Quelle zu bestimmen, sondern lediglich der Einfallswinkel θ, der für alle Mikrofone gleich ist.

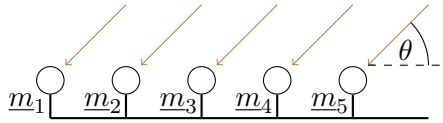


Abbildung 2.2: Im Fernfeldmodell ist der Abstand der Quelle um ein Vielfaches größer als der Durchmesser der Mikrofongruppe. Dann kann nur der Einfallswinkel θ geschätzt werden.

Im Gegensatz dazu ist das Nahfeldmodell mit einer sphärischen Wellenausbreitung genauer, weil die Einfallswinkel zu allen Mikrofonen unterschiedlich sind und deshalb die Quellposition in allen D Dimensionen genau bestimmt werden kann. Durch die Bestimmung aller D Koordinaten sind die Lokalisierungsverfahren, welche auf dem Nahfeldmodell basieren, prinzipiell aufwendiger in der Berechnung. Messungen haben gezeigt, dass in einem Büro mit einem Mikrofonarraydurchmesser von 1m das Nahfeld eine Gültigkeit bis 3m haben kann [Heilemann, 2012; Scheuing, 2007]. In dieser Arbeit werden die genauen Verfahren des Nahfelds betrachtet.

2.1.1 Reflexionsbehaftetes Signalmodell

Die TDOA-basierte Quellenlokalisierung benutzt lediglich die Eingangssignale $x_i(t)$ an den $1 \leq i \leq M$ Mikrofonen, welche sich aus der Faltung des Quellsignals $s(t)$ mit den Raumimpulsantworten $h_i(t)$ und einem Rauschen $\rho_i(t)$ nach

$$x_i(t) = h_i(t) * s(t) + \rho_i(t) \tag{2.1}$$

zusammensetzen. Die Raumimpulsantworten charakterisieren dabei die Raumeigenschaften wie Anzahl und Art der Reflexionen von der Quelle zu den Mikrofonen, wie sie in Abb. 2.3 skizziert sind.

In Abb. 2.4 ist eine solche Raumimpulsantwort aus einem Büroraum abgebildet. Die frühen Reflexionen zu den Zeitpunkten $t_{i,\mu}$ für das Mikrofon i und den Ausbreitungspfad μ sind innerhalb der ersten $0,03\,\mathrm{s}$ gut zu erkennen. Dabei stellt das erste lokale Maximum den Direktpfad dar. Es ist nicht immer das globale Maximum, weil sowohl die Abstrahlcharakteristik der Quelle und des Mikrofons als auch starke Reflexionen an Tischen zu einem weniger gedämpften Reflexionspfad führen können [Scheuing, 2007]. Den Reflexionen schließt sich der sogenannte Nachhall als Überlagerung vieler gedämpfter Reflexionen an.

Vereinfachend kann (2.1) als Überlagerung von verzögerten Wiederholungen des Quellsignals $s(t)$ zu den Zeiten $t_{i,\mu}$ mit einer Dämpfung $\alpha_{i,\mu}$ je Ausbreitungspfad μ angenommen werden. Dann ergibt sich mit P_i Ausbreitungspfaden je Mikrofon i das refle-

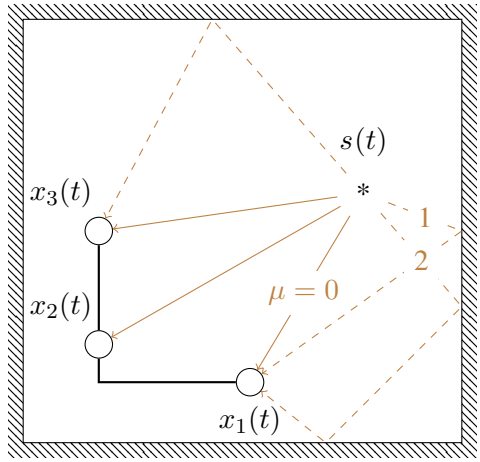


Abbildung 2.3: Räumliche Ausbreitung des Quellsignals $s(t)$ über mehrere Ausbreitungspfade μ. Die Mikrofone messen die Überlagerung als das Signal $x_i(t)$.

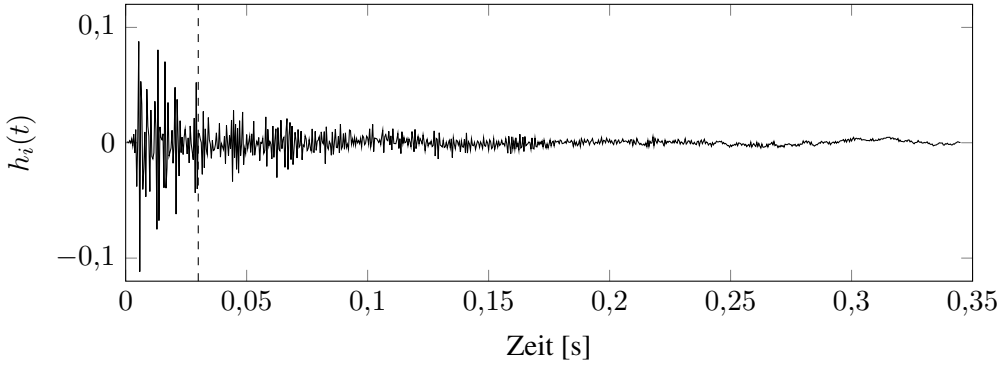


Abbildung 2.4: Gemessene Raumimpulsantwort $h_i(t)$ in einem Büroraum. Die frühen Reflexionen sind als Maxima zu erkennen gefolgt vom Nachhall.

xionsbehaftete Signalmodell

$$x_i(t) = \sum_{\mu=0}^{P_i-1} \alpha_{i,\mu} s(t - t_{i,\mu}) + \rho_i(t). \tag{2.2}$$

Der Nachhall sowie späte Reflexionen werden in diesem Modell vernachlässigt. Da das Quellsignal durch viele Reflexionen und dementsprechend lange Ausbreitungswege gedämpft und verzögert wird, erhält man einen abfallenden Verlauf der Impulsantwort. Ein Maß hierfür ist die Nachhallzeit T_{60}, welche den Abfall der maximalen Amplitude um 60dB beschreibt [Vorländer u. Mechel, 2002]. Diese ist in Abb. 2.5 für die Impulsantwort aus Abb. 2.4 mit einer relativ langen T_{60}-Zeit von 473ms gezeigt. Wie in

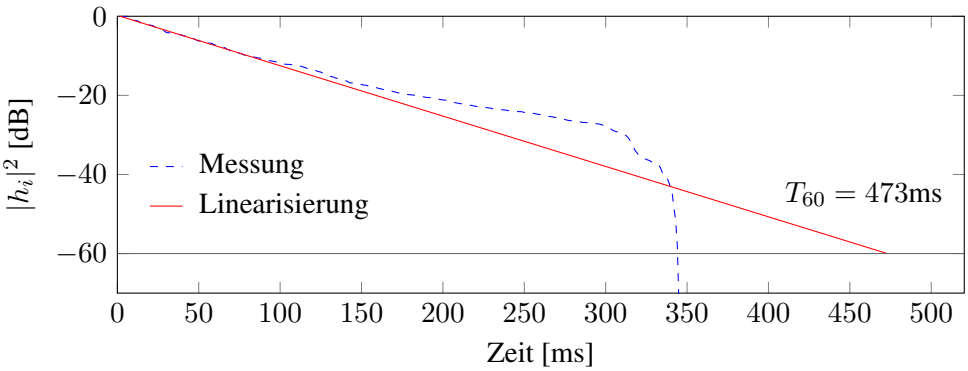


Abbildung 2.5: Zeitlicher Verlauf der Signalenergie $|h_i|^2$ aus Abb. 2.4. Die reale Messung wird linear angenähert.

[Loesch, 2013] diskutiert, ist die Bestimmung der T_{60}-Zeit nicht trivial, da sie entlang der Frequenzen unterschiedlich ist und einen nicht durchgehend linearen Abfall aufweist. Typischerweise nimmt man zur Bestimmung deshalb nur den Verlauf von -5dB bis -35dB und extrapoliert diese Differenz linear auf -60dB. In Abb. 2.5 wurde die Abschätzung aus [Schroeder, 1965] angewendet.

2.1.2 Räumliche Lokalisierung aus Abstandsdifferenzen

In einer reflexionsarmen Umgebung ist der Direktpfad stark ausgeprägt und kann als globales Maximum aus der Raumimpulsantwort geschätzt werden. Für vier Mikrofone mit den TOA $t_{i,0}\,(1 \leq i \leq 4)$ kann die Quellposition als Schnittpunkt der Kugeln mit den Radien $r_i = t_{i,0} \cdot c_0$ bestimmt werden, wobei c_0 die Schallgeschwindigkeit ist. Weil in vielen Anwendungen der Sendezeitpunkt unbekannt ist, kann man nicht direkt durch eine Maximasuche in $x_i(t)$ die Laufzeit bestimmen. Stattdessen nutzt man die TDOA

$$t_{ij,\mu\nu} = t_{i,\mu} - t_{j,\nu}, \tag{2.3}$$

welche aus der Differenz der TOA berechnet werden. Im Folgenden wird vereinfachend von dem verrauschten Direktpfad-TDOA $\tau_{ij} = t_{ij,00} + \rho_{ij}$ für eine Quelle zwischen Mikrofon i und j ausgegangen. Die genannten Lokalisierungsverfahren bekommen die geschätzten TDOA-Werte $\hat{\underline{\tau}}_v = [\hat{\tau}_{12}, \hat{\tau}_{13}, \ldots, \hat{\tau}_{(M-1)M}]^T$ und die Mikrofonpositionen $\underline{m}_i$ als Eingangsparameter und schätzen somit die Position der Quelle. Daraus ergibt sich das Signalflussdiagramm aus Abb. 2.6.

Gemäß der Definition ist die Abstandsdifferenz d_{ij} gleich der Differenz der euklidischen Abstände

$$d_{ij}(\underline{p}) = ||\underline{m}_i - \underline{p}|| - ||\underline{m}_j - \underline{p}||. \tag{2.4}$$

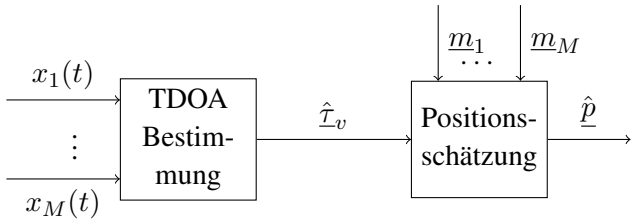


Abbildung 2.6: Verarbeitungskette zur TDOA-basierten Quellenlokalisierung

Durch die Messfehler in der TDOA-Schätzung ergibt sich die verrauschte Abstandsdifferenz

$$\delta_{ij} = \tau_{ij} c_0, \tag{2.5}$$

wobei c_0 die Schallgeschwindigkeit ist. Um die Position zu bestimmen, muss der Fehler

$$f(\underline{p}) = \sum_{1 \le i < j \le M} [\delta_{ij} - d_{ij}(\underline{p})]^2 \tag{2.6}$$

minimiert werden. Hierzu werden in der statistischen Signalverarbeitung Verfahren wie Maximum Likelihood (ML) oder Kleinste Quadrate (Least Squares, LS) verwendet [Köhler, 2005]. Unter der Annahme, dass die Fehler ρ_{ij} normalverteilte Rauschprozesse sind [Hahn u. Tretter, 1973; Huang u. a., 2004], ergibt sich die Wahrscheinlichkeitsdichtefunktion (WDF) bezüglich der Quellposition $\underline{p}$ zu

$$p(\underline{\delta}|\underline{p}) = \frac{\exp\{-\frac{1}{2}[\underline{\delta} - \underline{d}(\underline{p})]^T \mathbf{C}^{-1} [\underline{\delta} - \underline{d}(\underline{p})]\}}{\sqrt{(2\pi)^N \det(\mathbf{C})}} \tag{2.7}$$

mit der Kovarianzmatrix $\mathbf{C}$,

$$\underline{\delta} = [\delta_{12}, \ldots, \delta_{(M-1)M}]^T, \qquad \underline{d} = [d_{12}, \ldots, d_{(M-1)M}]^T \tag{2.8}$$

und $N = \frac{M(M-1)}{2}$ unterschiedlichen Mikrofonpaaren. Um eine geschlossene Lösung zu erhalten, setzen Verfahren wie [Smith u. Abel, 1987] und [Huang u. a., 2000, 2001] das Mikrofon $i = 1$ aus (2.4) als Referenz in den Koordinatenursprung $\underline{0}$. Anstelle aller N Mikrofonpaare nutzen diese Verfahren nur die $M-1$ sphärischen Mikrofonpaare, die mit dem Mikrofon $\underline{m}_1 = \underline{0}$ verbunden sind. Die sphärische TDOA-Menge wird auch als nicht-redundante TDOA-Menge bezeichnet im Gegensatz zur vollen bzw. redundanten TDOA-Menge.

Da bei den geschlossenen Verfahren, die auf der sphärischen TDOA-Menge basieren, viele TDOA nicht berücksichtigt werden, kann es zu einer Verschlechterung der Positionsschätzung kommen [Yang u. Scheuing, 2006]. Deshalb werden im Folgenden die Ansätze mit der sphärischen und der vollen TDOA-Menge separat betrachtet und in Kap. 6.3.3 bezüglich ihrer Genauigkeit und Rechenzeit verglichen.

2.1.3 Lokalisierung mit der sphärischen TDOA-Menge

Allgemeines Verfahren

Durch $\underline{m}_1 = \underline{0}$ in (2.4) ergibt sich

$$d_{1j}(\underline{p}) = d_j = ||\underline{p}|| - ||\underline{m}_j - \underline{p}|| \quad , \; j = 2, \ldots, M. \tag{2.9}$$

Durch Umsortieren und Quadrieren erhält man

$$(||\underline{p}|| - d_j)^2 = ||\underline{m}_j - \underline{p}||^2, \tag{2.10}$$

welches sich zu

$$||\underline{p}||^2 - 2d_j||\underline{p}|| + d_j^2 = ||\underline{m}_j||^2 - 2\underline{m}_j^T\underline{p} + ||\underline{p}||^2 \tag{2.11}$$

und

$$\underline{m}_j^T\underline{p} - d_j||\underline{p}|| = \underbrace{\frac{||\underline{m}_j||^2 - d_j^2}{2}}_{=b_j} \tag{2.12}$$

auflösen lässt [Huang u. a., 2000]. Werden statt der exakten Abstandsdifferenzen d_j die gemessenen Abstandsdifferenzen $\delta_{1j} = \tau_{1j}c_0$ eingesetzt, ist (2.12) nur noch näherungsweise erfüllt und es kann die Quellposition mit der LS-Methode bestimmt werden. Dafür formuliert man den Fehler in Abhängigkeit von der unbekannten Quellposition $\underline{p}$ und $||\underline{p}||$ als

$$f_{\mathrm{ULS}}(\underline{p}) = \sum_{j=2}^{M}(\underline{m}_j^T\underline{p} - \delta_j||\underline{p}|| - b_j)^2 \tag{2.13}$$

und mit

$$\mathbf{P} = \begin{bmatrix} \underline{m}_2^T & -\delta_2 \\ \vdots & \vdots \\ \underline{m}_M^T & -\delta_M \end{bmatrix} = \begin{bmatrix} \mathbf{M} & -\underline{d} \end{bmatrix}, \quad \underline{\theta} = \begin{bmatrix} \underline{p} \\ ||\underline{p}|| \end{bmatrix} \quad \text{und} \quad \underline{b} = \begin{bmatrix} b_2 \\ \vdots \\ b_M \end{bmatrix} \tag{2.14}$$

bekommt man

$$f_{\mathrm{ULS}}(\underline{\theta}) = ||\mathbf{P}\underline{\theta} - \underline{b}||^2. \tag{2.15}$$

Durch die separate Betrachtung des Betrags der Quellposition $||\underline{p}||$ kann (2.13) somit linearisiert werden und ist geschlossen lösbar. Gesucht ist folglich der unbeschränkte Kleinste Quadrate (Unconstrained Least Squares, ULS) Fehler

$$\hat{\underline{\theta}} = \arg\min_{\underline{\theta}} f_{\mathrm{ULS}}(\underline{\theta}), \tag{2.16}$$

der die geschätzte Quellposition liefert. Dieses konvexe Optimierungsproblem kann mittels der Pseudoinversen von $\mathbf{P}$ gelöst werden [Boyd u. Vandenberghe, 2004]. Hierzu muss $\mathrm{Rang}(\mathbf{P}) = D + 1$ und somit $M \geq D + 2$ sein. Gleichung (2.15) wird somit durch

$$\hat{\underline{\theta}} = \underbrace{(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T}_{=\mathbf{P}^+}\underline{b} \tag{2.17}$$

gelöst. Schlussendlich ist die geschätzte Quellposition mit

$$\hat{\underline{p}}_{\mathrm{ULS}} = \begin{bmatrix}\mathbf{I} & \underline{0}\end{bmatrix}\hat{\underline{\theta}} \tag{2.18}$$

gegeben. Es handelt sich hierbei um eine geschlossene Lösung des Lokalisierungsproblems, welche bis auf die Invertierungsoperation effizient erfolgt.

Vorangegangene Verfahren [Smith u. Abel, 1987; Friedlander, 1987] minimieren (2.13) über einen zweistufigen Ansatz. Dabei wird zuerst $f_{\mathrm{ULS}}(\underline{p})$ hinsichtlich $||\underline{p}||$ gelöst und anschließend $\underline{p}$ bestimmt. Ein weiterer Ansatz ist die Abbildung von (2.17) auf die zugehörigen Unterräume $R(\underline{p}) = \mathbb{R}^D$ und $R(r) = \mathbb{R}$. Diese lassen sich über eine blockweise Invertierung von $\mathbf{P}^+$ bestimmen als

$$\begin{aligned}\underline{p} = (\mathbf{QM})^+\,\underline{b} \quad \text{mit} \quad \mathbf{Q} = \mathbf{I} - \frac{\underline{d}\underline{d}^T}{||\underline{d}||^2} = \mathbf{I} - \frac{\underline{\tau}_s\underline{\tau}_s^T}{||\underline{\tau}_s||^2} \\ \text{und} \quad \underline{\tau}_s = [\tau_{12}, \ldots, \tau_{1M}]^T,\end{aligned} \tag{2.19}$$

der sphärischen TDOA-Menge und als

$$r = \frac{1}{c_0}\,(\mathbf{R}\underline{\tau}_s)^+\,\underline{b} \quad \text{mit} \quad \mathbf{R} = \mathbf{I} - \mathbf{M}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T \tag{2.20}$$

[Smith u. Abel, 1987; Huang u. a., 2000; Annibale u. Rabenstein, 2010]. Das Lokalisierungsproblem

$$\hat{\underline{p}} = \arg\min_{\underline{p}} ||\mathbf{Q}(\mathbf{P}\underline{\theta} - \underline{b})||^2 \tag{2.21}$$

ist gleichbedeutend zu (2.13) und kann mittels

$$\hat{\underline{p}} = (\mathbf{M}^T\mathbf{QM})^{-1}\mathbf{M}^T\mathbf{Q}\underline{b} \tag{2.22}$$

gelöst werden [Stoica u. Li, 2006; Huang u. a., 2001].

Allgemeines Verfahren mit Nebenbedingung

In $\underline{\theta}$ aus (2.17) wird der Zusammenhang zwischen den euklidischen Koordinaten der Quellposition $\underline{p} = [p_1, \ldots, p_D]^T$ und dem euklidischen Abstand der Quelle $||\underline{p}|| =$

$\sqrt{\sum_{\iota=1}^{D} p_\iota^2}$ nicht beachtet. Diese Nebenbedingung kann in die Positionsbestimmung durch

$$\underline{\theta}^T \mathbf{D} \underline{\theta} = 0, \quad \mathbf{D} = \begin{pmatrix} \mathbf{I} & \underline{0} \\ \underline{0}^T & -1 \end{pmatrix} \tag{2.23}$$

und den Lagrange-Multiplikator λ mit einfließen:

$$f_{\mathrm{CLS}}(\underline{p}, \lambda) = ||\mathbf{P}\underline{\theta} - \underline{b}||^2 + \lambda \cdot \underline{\theta}^T \mathbf{D} \underline{\theta}. \tag{2.24}$$

Dieser Ansatz wird in dieser Arbeit als beschränkte Kleinste Quadrate (Constrained Least Squares, CLS) Methode bezeichnet. Um f_{CLS} zu minimieren, wird in der Literatur eine näherungsweise Lösung mittels Lagrange Entwicklung angegeben [Stoica u. Li, 2006; Huang u. a., 2001]. Dabei wird mit Erweitern

$$f_{\mathrm{CLS}}(\underline{p}, \lambda) = \underline{\theta}^T (\mathbf{P}^T \mathbf{P} + \lambda \mathbf{D}) \underline{\theta} - 2\underline{b}^T \mathbf{P} \underline{\theta} + \underline{b}^T \underline{b} \tag{2.25}$$

und durch $\partial f_{\mathrm{CLS}}(\underline{p}, \lambda) / \partial \underline{\theta} = \underline{0}$ die LS Lösung

$$\hat{\underline{\theta}}(\lambda) = (\mathbf{P}^T \mathbf{P} + \lambda \mathbf{D})^{-1} \mathbf{P}^T \underline{b} \tag{2.26}$$

mit unbekanntem λ abgeleitet. Hierbei muss berücksichtigt werden, dass $(\mathbf{P}^T \mathbf{P} + \lambda \mathbf{D})$ positiv definit ist. Den passenden Lagrange-Multiplikator λ_0 erhält man schließlich durch erneutes Einsetzen in die Nebenbedingung

$$\hat{\underline{\theta}}^T (\lambda_0) \mathbf{D} \hat{\underline{\theta}}(\lambda_0) = 0. \tag{2.27}$$

Diese Gleichung kann mit dem iterativen Gradientenverfahren gelöst werden [Bronstein u. Semendjajew, 2008]. Aus dem ermittelten λ_0 erfolgt die Positionsberechnung mit

$$\hat{\underline{p}}_{\mathrm{CLS}} = \begin{bmatrix} \mathbf{I} & \underline{0} \end{bmatrix} \hat{\underline{\theta}}(\lambda_0). \tag{2.28}$$

2.1.4 Lokalisierung mit der vollen TDOA-Menge

Im Gegensatz zu den bisherigen Lokalisierungsverfahren, die auf der sphärischen TDOA-Menge $\underline{\tau}_s$ basieren, werden in diesem Kapitel Verfahren diskutiert, welche die volle TDOA-Menge $\underline{\tau}_v$ nutzen. Geht man – wie weithin angenommen [Hahn u. Tretter, 1973; Chan u. Ho, 1994b] – bei den Fehlern ρ_{ij} von unabhängigen und mittelwertfreien gaußschen Rauschprozessen aus, kann für (2.7) ein ML-Schätzverfahren angegeben werden. Um die Position zu finden, muss die WDF $p(\underline{\delta}|\underline{p})$ maximiert werden. Da die Exponentialfunktion streng monoton steigend ist, kann ebenso

$$f_{\mathrm{ML}}(\underline{p}) = -\ln p(\underline{\delta}|\underline{p}) \tag{2.29}$$

$$= -\frac{1}{2}[\underline{\delta} - \underline{d}(\underline{p})]^T \mathbf{C}^{-1} [\underline{\delta} - \underline{d}(\underline{p})] - \frac{1}{2} \ln\left((2\pi)^N \det(\mathbf{C})\right) \tag{2.30}$$

minimiert werden. Somit folgt das Optimierungskriterium direkt als

$$\underline{\hat{p}}_{\mathrm{ML}} = \arg\min_{\underline{p}} f_{\mathrm{ML}}(\underline{p}). \tag{2.31}$$

Aufgrund der nichtlinearen Problemstellung existiert kein wirksamer Schätzer für $\underline{\hat{p}}$ [Yang u. Scheuing, 2005]. Dennoch führen die zusätzlichen TDOA-Werte der vollen TDOA-Menge im Vergleich zur sphärischen TDOA-Menge zu einer genaueren Positionsschätzung [Yang u. Scheuing, 2006].

Unter der Annahme von unabhängigen Fehlern ist $\mathbf{C}$ eine Diagonalmatrix mit den Varianzen σ_n^2 der jeweiligen TDOA-Fehler $\mathbf{C} = \mathrm{diag}\left(\sigma_1^2, \ldots, \sigma_N^2\right)$. Dies führt zum Optimierungskriterium der gewichteten Kleinste Quadrate (Weighted Least Squares, WLS)

$$f_{\mathrm{WLS}}(\underline{p}) = \sum_{1 \le i < j \le M} \frac{[\delta_{ij} - d_{ij}(\underline{p})]^2}{\sigma_{ij}^2}. \tag{2.32}$$

Dieses Verfahren ist eine Erweiterung des LS-Kriterium aus (2.6).

Gauß-Newton-Verfahren

Um das Minimum einer N-dimensionalen, nichtlinearen Funktion $f(\underline{x})$ zu finden, werden häufig iterative Verfahren angewendet, die sich von einem Startwert $\underline{x}_0$ in mehreren Schritten dem Minimum nähern. Dabei muss die Schrittweite $\underline{\chi}_k$ so gefunden werden, dass $\underline{x}_{k+1} = \underline{x}_k + \underline{\chi}_k$ die Funktion bestmöglich minimiert. Dabei wird die Funktion f durch eine Taylor-Reihe 2. Ordnung angenähert

$$f(\underline{x}_k + \underline{\chi}_k) \approx f(\underline{x}_k) + \underline{\chi}_k^T \underline{g}_k + \frac{1}{2} \underline{\chi}_k^T \mathbf{H}_k \underline{\chi}_k, \tag{2.33}$$

mit dem Gradienten $\underline{g}_k = \left[\frac{\partial f(\underline{x}_k)}{\partial x_1}, \ldots, \frac{\partial f(\underline{x}_k)}{\partial x_D}\right]^T$ und der Hesse-Matrix

$$\mathbf{H}_k = \begin{bmatrix} \frac{\partial f^2(\underline{x}_k)}{\partial x_1 \partial x_1} & \frac{\partial f^2(\underline{x}_k)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial f^2(\underline{x}_k)}{\partial x_1 \partial x_D} \\ \frac{\partial f^2(\underline{x}_k)}{\partial x_2 \partial x_1} & \frac{\partial f^2(\underline{x}_k)}{\partial x_2 \partial x_2} & \cdots & \frac{\partial f^2(\underline{x}_k)}{\partial x_2 \partial x_D} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f^2(\underline{x}_k)}{\partial x_D \partial x_1} & \frac{\partial f^2(\underline{x}_k)}{\partial x_D \partial x_2} & \cdots & \frac{\partial f^2(\underline{x}_k)}{\partial x_D \partial x_D} \end{bmatrix}$$

zum Iterationsschritt k. Um das Minimum von f zu bestimmen, berechnet man die Ableitung $\partial f(\underline{x}) / \partial \underline{\chi}_k$ aus (2.33) und setzt diese Null. Daraus ergibt sich

$$\underline{g}_k + \mathbf{H}_k \underline{\chi}_k = 0 \quad \text{und} \quad \underline{\chi}_k = -\mathbf{H}_k^{-1} \underline{g}_k. \tag{2.34}$$

Aus dieser Schrittweite lässt sich die verbesserte Schätzung des Argumentes

$$\underline{x}_{k+1} = \underline{x}_k - \alpha \mathbf{H}_k^{-1} \underline{g}_k \tag{2.35}$$

bestimmen, mit dem Koeffizienten α, welcher die Funktion minimiert. Der Fall $\alpha = 1$ gilt, wenn (2.33) keine Näherung und $\mathbf{H}$ regulär ist. Diese Lösung beschreibt das Newton Verfahren [Boyd u. Vandenberghe, 2004]. Handelt es sich bei

$$f(\underline{x}) = \underline{f}^T(\underline{x})\underline{f}(\underline{x}) = \sum_{n=1}^{N} f_n^2(\underline{x}) \tag{2.36}$$

um einen quadratischen Fehler, ergibt sich hieraus der Spezialfall des Gauß-Newton (GN)-Verfahrens [Antoniou u. Lu, 2007]. Dabei beschreibt

$$\frac{\partial f(\underline{x})}{\partial x_d} = \sum_{n=1}^{N} 2 f_n(\underline{x}) \frac{\partial f_n(\underline{x})}{\partial x_d} \tag{2.37}$$

das d-te Element des Gradientenvektors und wird in Matrix-Form zu

$$\underline{g} = 2\mathbf{J}^T \underline{f}(\underline{x}) \tag{2.38}$$

mit der Jakobi-Matrix

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_D} \\ \vdots & & & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \cdots & \frac{\partial f_N}{\partial x_D} \end{bmatrix} \tag{2.39}$$

kompakt dargestellt. Um die zweite Ableitung für die Taylor-Reihe zu erhalten, muss (2.37) erneut abgeleitet werden und man erhält

$$\frac{\partial^2 f}{\partial x_d \partial x_e} = 2 \sum_{n=1}^{N} \frac{\partial f_n}{\partial x_d} \frac{\partial f_n}{\partial x_e} + 2 \sum_{n=1}^{N} f_n \cdot \frac{\partial^2 f_n}{\partial x_d \partial x_e}. \tag{2.40}$$

Unter Vernachlässigung der 2. Ableitungen, bekommt man

$$\frac{\partial^2 f}{\partial x_d \partial x_e} \approx 2 \sum_{n=1}^{N} \frac{\partial f_n}{\partial x_d} \frac{\partial f_n}{\partial x_e} \tag{2.41}$$

und damit eine Abschätzung für die Hesse-Matrix

$$\mathbf{H} \approx 2\mathbf{J}^T\mathbf{J}. \tag{2.42}$$

Zum Minimieren von (2.36) berechnet sich die schrittweise Annäherung an das Argument äquivalent zu (2.35) geschlossen als

$$\underline{x}_{k+1} = \underline{x}_k - \alpha_k (\mathbf{J}^T\mathbf{J})^{-1} (\mathbf{J}^T \underline{f}). \tag{2.43}$$

Für den Spezialfall der Lokalisierung ist $\underline{x}_k = \underline{p}_k$ und $f(\underline{x}_k) = f_{\mathrm{WLS}}(\underline{p}_k)$ aus (2.32). Mit $f_n(\underline{p}_k) = (\delta_{ij} - d_{ij}(\underline{p}_k))/\sigma_{ij}$ für alle $1 \leq i < j \leq M$ Mikrofonpaare erhält man die Jakobi-Matrix

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1(\underline{p}_k)}{\partial p_1} & \frac{\partial f_1(\underline{p}_k)}{\partial p_2} & \frac{\partial f_1(\underline{p}_k)}{\partial p_3} \\ \vdots & & \vdots \\ \frac{\partial f_N(\underline{p}_k)}{\partial p_1} & \frac{\partial f_N(\underline{p}_k)}{\partial p_2} & \frac{\partial f_N(\underline{p}_k)}{\partial p_3} \end{bmatrix} = \sqrt{\mathbf{C}}^{-1} \cdot \bar{\mathbf{J}} \tag{2.44}$$

$$\text{mit} \quad \sqrt{\mathbf{C}}^{-1} = \mathrm{diag}\left(\frac{1}{\sigma_{12}}, \ldots, \frac{1}{\sigma_{(M-1)M}}\right), \quad \bar{\mathbf{J}} = \begin{bmatrix} (\underline{u}_2 - \underline{u}_1)^T \\ \vdots \\ (\underline{u}_M - \underline{u}_{M-1})^T \end{bmatrix}$$

$$\text{und} \quad \underline{u}_m = \frac{\underline{p}_k - \underline{m}_m}{||\underline{p}_k - \underline{m}_m||}, \ (m = 1, \ldots, M)$$

[Chan u. Ho, 1994a; So u. a., 2008]. Nach [Foy, 1976] ergibt sich somit für (2.32) der Verbesserungsterm je Schritt des GN-Verfahrens zu

$$\underline{\chi}_k = (\bar{\mathbf{J}}^T \mathbf{C}^{-1} \bar{\mathbf{J}})^{-1} \bar{\mathbf{J}}^T \mathbf{C}^{-1} (\underline{\delta} - \underline{d}(\underline{p}_k)). \tag{2.45}$$

Die Positionsschätzung mittels des GN Verfahrens ist

$$\hat{\underline{p}}_{\mathrm{GN}} = \underline{p}_k - \underline{\chi}_k, \tag{2.46}$$

sobald f_{WLS} mit einer geschätzten Position $\underline{p}_k$ einen Grenzwert unterschreitet oder eine maximale Anzahl von Schritten erreicht ist. Probleme des GN-Verfahrens sind die Initialisierung und daraus die Gefahr, in ein lokales Minimum zu gelangen. Außerdem ist die Konvergenz ein nicht beeinflussbarer Faktor [Huang u. a., 2001].

Optimale Abbildung der TDOA

Die TDOA besitzen die Eigenschaft der Redundanz, da

$$t_{ik} + t_{kj} = t_i - t_k + t_k - t_j = t_i - t_j = t_{ij} \tag{2.47}$$

gilt [Schmidt, 1996; So u. a., 2008]. Daraus lässt sich ableiten, dass die TDOA-Werte der vollen TDOA-Menge $t_v = [t_{12}, \ldots, t_{(M-1)M}]^T$ eine Linearkombination der sphärischen TDOA-Menge $t_s = [t_{12}, \ldots, t_{1M}]^T$ sind. Es existiert somit eine lineare Abbildung für die rauschfreien TDOA-Werte

$$t_v = \mathbf{B} t_s, \tag{2.48}$$

welche jede TDOA der vollen TDOA-Menge als Differenz zweier TDOA der sphärischen TDOA-Menge darstellt:

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_{M-1} \\ \tilde{\mathbf{B}} \end{bmatrix}, \ \tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{X}_2 \\ \mathbf{X}_3 \\ \vdots \\ \mathbf{X}_{M-1} \end{bmatrix} \text{ und } \mathbf{X}_i = \begin{bmatrix} \underline{0}_{(M-i)\times(i-2)} & -\underline{1}_{(M-i)} & \mathbf{I}_{M-i} \end{bmatrix}. \tag{2.49}$$

Dabei ist $\mathbf{B}$ eine $N \times (M-1)$ und $\mathbf{X}_i$ eine $(M-i) \times (M-1)$ Matrix und $\underline{1}$ ein Spaltenvektor mit nur Eins-Elementen. Da $\mathbf{B}$ einen vollen Spaltenrang von $M-1$ besitzt, kann

$$t_s = \mathbf{B}^+ t_v \tag{2.50}$$

berechnet werden. In [So u. a., 2008] ist hierfür der geschlossene Ausdruck

$$\mathbf{B}^+ = \frac{1}{M} \begin{bmatrix} \mathbf{I}_{M-1} + \underline{1}_{M-1}\underline{1}_{M-1}^T & \tilde{\mathbf{B}}^T \end{bmatrix} \tag{2.51}$$

gegeben.

Für fehlerhafte TDOA-Schätzungen ist (2.48) nicht mehr eindeutig und kann nur noch näherungsweise erfüllt werden. Dementsprechend ist es Ansatz für die LS-Verfahren aus Kap. 2.1.3, die verrauschten, sphärischen TDOA-Werte in $\underline{\tau}_s$ durch die TDOA-Schätzungen aus $\mathbf{B}^+\underline{\tau}_v$ zu verbessern. Dabei stellt $\mathbf{B}^+$ ein überbestimmtes Gleichungssystem dar und die resultierenden TDOA-Werte $\underline{\tau}_s$ ergeben sich als eine Mittelung der Werte aus $\underline{\tau}_v$. In Kap. 6.3.3 werden die zwei Methoden anhand von Simulationen verglichen.

2.2 TDOA-Schätzung mit dem Korrelationsverfahren

Zur Schätzung der TDOA wird die Ähnlichkeit in den Mikrofonsignalen gesucht, um damit den zeitlichen Versatz des Quellsignals zu bestimmen [Carter, 1987]. Diese lässt sich mit der Kreuzkorrelation

$$r_{ij}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x_i(t) \cdot x_j(t+\tau) dt \tag{2.52}$$

berechnen [Vary u. a., 1998]. Die Kreuzkorrelation liefert für zwei stationäre und ergodische Rauschsignale $x_i(t)$ und $x_j(t) = x_i(t-\tau_0)$, die sich nur um den Versatz τ_0 unterscheiden, eine Korrelationsfunktion mit dem Maximum bei $\tau = \tau_0$. Um die Kreuzkorrelation auf unterschiedliche Signale und Szenarien anzupassen, haben [Knapp

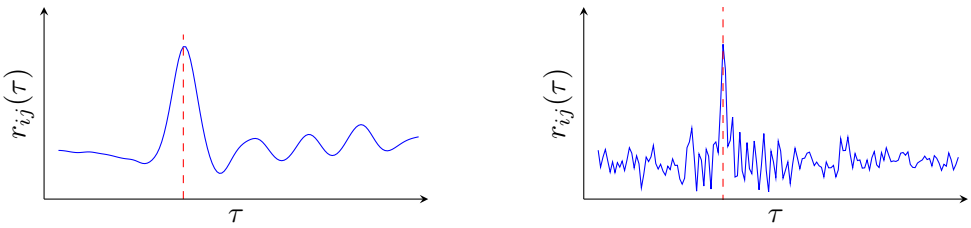
u. Carter, 1976] die verallgemeinerte Kreuzkorrelation (Generalized Cross-Correlation, GCC) eingeführt, welche über einen Vorfilter $\Psi(\omega)$ im Frequenzbereich effizient implementiert werden kann. Die GCC berechnet sich nach

$$r_{ij}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Psi(\omega) X_i(\omega) X_j^*(\omega)\, \mathrm{e}^{\mathrm{j}\omega\tau}\, d\omega, \tag{2.53}$$

inklusive der inversen Fourier-Transformation und den Fourier-transformierten Mikrofonsignalen $X_i(\omega)$ und $X_j(\omega)$. Bekannte Vorfilter sind die von Roth [Roth, 1971] oder SCOT [Carter u. a., 1973]. Allerdings hat sich gezeigt, dass der Phasentransformationsfilter (PHAT)

$$\Psi_{\mathrm{PHAT}}(\omega) = \frac{1}{|X_i(\omega) X_j^*(\omega)|} \tag{2.54}$$

mit einer Normierung im Frequenzbereich (*whitening*) in moderaten echobehafteten Umgebungen die besten Ergebnisse liefert [Dibiase u. a., 2001]. Da im Bereich der akustischen Lokalisierung breitbandige Signale vorliegen, sollen alle Frequenzen gleich gewichtet werden. Ein Nachteil des PHAT Filters ist, dass in stark hallenden Umgebungen oder bei geringem Signal-zu-Rauschabstand (signal-to-noise-ratio, SNR) auch die Rauschkomponenten verstärkt werden.



(a) GCC mit $\Psi(\omega) = 1$ (b) GCC mit $\Psi(\omega) = \Psi_{\mathrm{PHAT}}(\omega)$

Abbildung 2.7: Vergleich der allgemeinen Kreuzkorrelation und der GCC-PHAT für zwei reale Mikrofonsignale für eine Rauschquelle bei $T_{60} \approx 80\mathrm{ms}$

Dies veranschaulicht Abb. 2.7, in der die allgemeine Kreuzkorrelation und die GCC mit dem PHAT Vorfilter

$$r_{ij}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{X_i(\omega) X_j^*(\omega)}{|X_i(\omega) X_j^*(\omega)|}\, \mathrm{e}^{\mathrm{j}\omega\tau}\, d\tau \tag{2.55}$$

für eine Quelle in einem Büroraum gezeigt sind. In der GCC-PHAT ist ein schmaleres Maximum an der Stelle der TDOA von Mikrofon i zu Mikrofon j zu erkennen. Es ist dadurch eine genauere Auflösung entstanden, allerdings auf Kosten von mehr Nebenmaxima.

Aus der GCC-PHAT im Zeitbereich lassen sich mit einer Maximasuche

$$\hat{\tau}_{ij} = \arg\max_{\tau} \; r_{ij}(\tau) \tag{2.56}$$

die TDOA $\hat{\tau}_{ij}$ schätzen. Weitere Verfahren wie die gemeinsame Schätzung aller TDOA über eine Eigenwertzerlegung der Kovarianzmatrix der Mikrofonsignale [Jacob, 2000] sind aufgrund ihres erhöhten Rechenaufwands für diese Arbeit nicht praktikabel.

2.3 Lokalisierung mehrerer Quellen

Nach den Verfahren der Einzelquellenlokalisierung betrachtet dieses Kapitel die Probleme und Ansätze für eine simultane Mehrquellenlokalisierung.

2.3.1 Mehrdeutigkeiten in der TDOA-Schätzung

In Abb. 2.8 ist eine GCC-PHAT von zwei Quellen in einer realen Umgebung abgebildet. Es ist deutlich zu erkennen, dass das Maximum der zweiten Quelle nicht ausgeprägt ist. Dieser Effekt hat mehrere Ursachen.

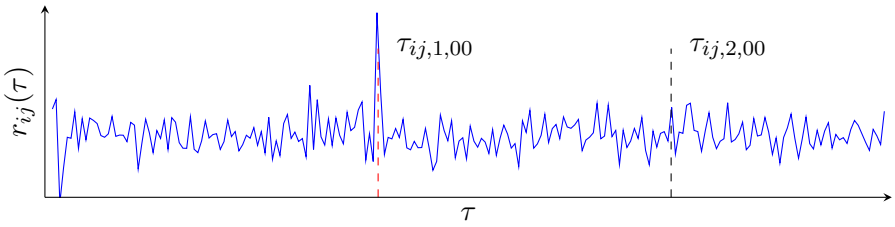


Abbildung 2.8: GCC-PHAT in einem echobehafteten Raum mit $T_{60} \approx 600$ms und für zwei Sprachquellen

Wie bereits erwähnt, führen zum einen die Reflexionspfade μ zu Verzögerungen $t_{iq,\mu}$ des gleichen Quellsignals $s_q(t)$ mit einer Dämpfung $\alpha_{iq,\mu}$. Sind mehrere Quellen $1 \leq q \leq Q$ simultan aktiv, erweitert sich das lineare Modell aus (2.2) zu

$$x_i(t) = \sum_{q=1}^{Q} \sum_{\mu=0}^{P_{iq}-1} \alpha_{iq,\mu} s_q(t - t_{iq,\mu}) + \rho_i(t). \tag{2.57}$$

Aus der GCC-PHAT erhält man somit an jedem Mikrofonpaar (i, j) einen Peak bei

$$\tau_{ij,q,\mu\nu} = t_{iq,\mu} - t_{jq,\nu} \tag{2.58}$$

für jede Quelle q mit den zugehörigen Ausbreitungspfaden μ und ν ($0 \leq \mu \leq P_{iq} - 1$ und $0 \leq \nu \leq P_{jq} - 1$). Zusätzlich führen Periodizitäten in akustischen Signalen wie Sprache und Musik zu weiteren Maxima der Kreuzkorrelation.

Alle genannten Punkte führen zu einer uneindeutigen Zuordnung der Direktpfad-Maxima an einem Mikrofonpaar (i, j) zu einer Quelle q. Außerdem ist neben der Wahl der Direktpfade besonders die Kombination der zugehörigen TDOA für eine Quelle entlang der unterschiedlichen Mikrofonpaare nicht trivial, da die Peaks in der GCC-PHAT soweit keine Rückschlüsse auf ihre Quelle erlauben. Im Weiteren dieser Arbeit werden die uneindeutig zugeordneten TDOA-Schätzungen mit $\hat{\tau}_{ij}$ abgekürzt, wobei je Mikrofonpaar K_{ij} TDOA-Schätzungen betrachtet werden. Dabei muss $K_{ij} \geq Q$ gewährleistet sein.

2.3.2 Überprüfung aller TDOA-Kombinationen

Um die richtigen Direktpfad-TDOA-Kombinationen für die Mehrquellenlokalisierung zu finden, wird die Verarbeitungskette aus Abb. 2.6 um einen Zwischenschritt erweitert, siehe Abb. 2.9. Es werden dabei alle $\prod_{1 \leq i < j \leq M} K_{ij}$ TDOA-Kombinationen überprüft, wobei sehr viele TDOA-Kombinationen entstehen, die keiner realen Quelle zuzuordnen sind. Deshalb ist es notwendig, ein Kriterium für die Validierung der TDOA-Kombinationen zu finden und diese zuverlässig einer Quelle zuzuweisen.

Abbildung 2.9: Flussdiagramm zur indirekten Sprecherlokalisierung

Schallgeschwindigkeitskriterium In [Hu u. Yang, 2010] wird die TDOA-basierte Schallgeschwindigkeitsschätzung als Kriterium vorgeschlagen. Dabei legen die Autoren eine planare Schallwelle (Fernfeld) zugrunde und schätzen aus jeder TDOA-Kombination $\underline{\hat{\tau}}_s = [\hat{\tau}_{12}, \ldots, \hat{\tau}_{1M}]^T$ die Schallgeschwindigkeit $\hat{c}$. Liegt dieser geschätzte Wert nahe der echten Schallgeschwindigkeit c_0, d. h.

$$|\hat{c} - c_0| < e_c, \tag{2.59}$$

so wird die TDOA-Kombination $\underline{\hat{\tau}}_s$ als richtig akzeptiert. Der Parameter e_c ist der Schwellwert der akzeptierten Abweichung der Schallgeschwindigkeit. Dieses Verfahren hat zwei Nachteile:

1. Die Fernfeldannahme trifft meistens auf Innenraumanwendungen nicht zu.

2. Die Anzahl an getesteten TDOA-Kombinationen wächst exponentiell mit $\prod_{2\leq j\leq M} K_{1j}$

In [Annibale u. a., 2013b] haben die Autoren gezeigt, dass der Nahfeldansatz allgemein bessere Ergebnisse liefert. Deshalb wird er im Folgenden genauer erläutert.

Aus (2.19) und (2.20) kann die Abhängigkeit von der Schallgeschwindigkeit für die Quelle q als

$$\underline{p}_q(c) = \underbrace{(\mathbf{QM})^+}_{\mathbf{G}}\underline{b}(c) \quad \text{und} \quad r_q(c) = \frac{1}{c}\underbrace{(\mathbf{R}_{\underline{\tau}_s})^+}_{\underline{h}}\underline{b}(c) \tag{2.60}$$

formuliert werden [Annibale u. a., 2013a; Annibale u. Rabenstein, 2013]. Durch Gleichsetzen $||\underline{p}_q(c)||^2 = r_q^2(c)$ erhält man

$$c^2\underline{b}^T(c)\mathbf{G}^T\mathbf{G}\underline{b}(c) = \underline{b}(c)^T\underline{h}^T\underline{h}\underline{b}(c). \tag{2.61}$$

Ersetzt man $\underline{b}(c) = [b_2(c), \ldots, b_M(c)]^T$ mit $b_j(c) = (||\underline{m}_j||^2 - c^2\tau_{1j}^2)/2$ und substituiert $k = c^2$, erhält man

$$\underline{\beta}^T\mathbf{G}^T\mathbf{G}\underline{\beta}\,k^3 + (-\underline{\alpha}^T\mathbf{G}^T\mathbf{G}\underline{\beta} - \underline{\beta}^T\mathbf{G}^T\mathbf{G}\underline{\alpha} - \underline{\beta}^T\underline{h}^T\underline{h}\underline{\beta})\,k^2 + (\underline{\alpha}^T\mathbf{G}^T\mathbf{G}\underline{\alpha} + \underline{\alpha}^T\underline{h}^T\underline{h}\underline{\beta} + \underline{\beta}^T\underline{h}^T\underline{h}\underline{\alpha})\,k - \underline{\alpha}^T\underline{h}^T\underline{h}\underline{\alpha} = 0 \tag{2.62}$$

$$\text{mit} \quad \underline{\alpha} = \left[||\underline{m}_2||^2, \ldots, ||\underline{m}_M||^2\right]^T \text{ und } \underline{\beta} = \left[\tau_{12}^2, \ldots, \tau_{1M}^2\right]^T. \tag{2.63}$$

Nach Lösen dieses Polynoms 3. Ordnung ist eine Lösung für die Schallgeschwindigkeitsschätzung im Nahfeld gegeben. Dieses Kriterium wurde bereits erfolgreich in [Annibale u. a., 2013b] angewendet.

Kriterium des residualen Fehlers Eine weitere Möglichkeit, die TDOA-Kombinationen zu überprüfen, ist der residuale TDOA-Fehler aus [Scheuing u. Yang, 2008]. Dabei berechnet man aus der geschätzten Position $\hat{\underline{p}}$ die potentiellen TDOA-Werte $\tilde{\tau}_{ij} = \frac{1}{c_0}d_{ij}(\hat{\underline{p}})$ nach (2.4) je Mikrofonpaar. Vergleicht man diese Werte mit der geschätzten TDOA-Kombination $\hat{\tau}_{ij}$, so erhält man das zweite mögliche Fehlerkriterium

$$\frac{||\hat{\underline{\tau}} - \tilde{\underline{\tau}}||}{\sqrt{n}} < e_t. \tag{2.64}$$

Dabei ist n die Anzahl der verwendeten TDOA-Werte in $\hat{\underline{\tau}}$ und e_t der Schwellwert für eine akzeptierte TDOA-Kombination $\hat{\underline{\tau}}$.

Sowohl das Kriterium aus (2.59) als auch (2.64) werden in Kap. 6.3.4 anhand von Simulationen verlichen. Beide Kriterien helfen zwar, die falschen TDOA-Kombinationen zu finden und zu eliminieren, dennoch müssen alle $\prod_{2\leq j\leq M} K_{1j}$ Kombinationen geprüft werden, die exponentiell mit M ansteigen.

2.3.3 Auflösung der TDOA-Mehrdeutigkeiten

Dieser Ansatz stellt eine kombinatorische Optimierung dar, die mittels der Nullsumme von TDOA-Werten die möglichen TDOA-Kombinationen findet. Stammen die TDOA-Werte von derselben Quelle und von den gleichen Ausbreitungspfaden, so lässt sich entlang einer Masche von Mikrofonen $i, j, k, \ldots, l, i$ die Bedingung

$$\begin{aligned} \tau_{ij,q,\mu\nu} + \tau_{jk,q,\nu\kappa} + \cdots + \tau_{li,q,\eta\mu} = \\ t_{iq,\mu} - t_{jq,\nu} + t_{jq,\nu} - t_{kq,\kappa} + \cdots + t_{lq,\eta} - t_{iq,\mu} = 0 \end{aligned} \tag{2.65}$$

aufstellen. Diese Eigenschaft wurde bereits in [Schmidt, 1996] angewendet, um die TDOA entlang eines Mikrofon-Tripels robuster zu schätzen (*TDOA averaging*). In dem Verfahren DATEMM [Scheuing u. Yang, 2008] wird (2.65) auf Mikrofon-Tripel angewendet, um falsche Kombinationen zu detektieren. Dabei werden lediglich $\prod_n K_n$, $n \in \{ij, jk, ki\}$ TDOA-Kombinationen je Mikrofon-Tripel überprüft und diejenigen Kombinationen gespeichert, welche Null ergeben. Diese TDOA-Tripel heißen konsistent und werden anschließend zusammengefügt. Dafür wird ein Referenz-Tripel gewählt und sukzessiv alle weiteren konsistenten Tripel hinzugefügt, die gleiche TDOA auf gemeinsamen Kanten haben. Das Problem lässt sich als Graph abstrahieren, der als Knoten die Mikrofone und als Kantengewichte die TDOA-Werte besitzt. Wird der Graph mit den konsistenten Tripeln synthetisiert, ist er konsistent. Sind alle möglichen konsistenten Tripel zusammengefügt, so repräsentiert dieser synthetisierte, konsistente Graph eine mögliche Direktpfadkombination von TDOA-Werten. Im Anschluss wird nur für die reduzierte Anzahl von konsistenten Graphen mit den TDOA-Werten als Kantengewichte eine Lokalisierung durchgeführt.

Das DATEMM Verfahren bietet eine Methode zur robusten, simultanen Lokalisierung akustischer Quellen und wurde bereits erfolgreich in einem echtzeitfähigen System angewendet [Scheuing, 2007; Kaschub, 2007]. Allerdings weist DATEMM Optimierungspotential auf:

1. Verlust von Lösungen: Wenn ein Tripel in einem konsistenten Graphen verwendet wurde, wird es für die weitere Synthese ausgeschlossen, siehe [Kreißig u. Yang, 2011].
2. Feste Länge der Kombinationen: Die Wahl von Tripel als zyklisches Grundelement ist nicht immer die beste Wahl, da manche TDOA-Werte fehlen können und damit kein Tripel gebildet werden kann.
3. Die Konsistenzbedingung ist eine notwendige aber nicht hinreichende Bedingung für die Direktpfad-TDOA.

Diese Nachteile werden in Kap. 4.1.2 und Kap. 6.2.2 noch genauer analysiert und diskutiert. In Kap. 4.2 führen wir einen Algorithmus zur Kombination konsistenter TDOA-Kombinationen ein, der keine Lösungen verliert und anschließend führen wir den all-

gemeinen Algorithmus zur Synthese konsistenter Graphen ein, der mit einer beliebigen Kombination und Anzahl von TDOA-Werten arbeitet. In Kap. 6.3.4 stellen wir unterschiedliche Nachverarbeitungsschritte vor, mit denen die richtigen Direktpfad-TDOA aus den konsistenten Kombinationen selektiert werden können und vergleichen diese.

2.3.4 Lokalisierung mit dem ICA-Verfahren

Für die Lokalisierung mehrerer Quellen in echobehafteten Umgebungen sind neben den bisher behandelten korrelationsbasierten Verfahren auch statistische Verfahren bekannt, die u.a. die Raumimpulsantwort und dadurch auch die Position der Quellen schätzen. Hierzu zählt vor allem die Blinde Quellentrennung (blind source separation, BSS), welche eine Independent Component Analysis (ICA) anwendet, um die akustischen Signale zu trennen [Loesch u. a., 2010; Nesta u. Omologo, 2012a]. Die BSS bestimmt die Raumimpulsantworten und invertiert diese, um die reinen akustischen Quellsignale zu erhalten. Dabei wird das Modell von statistisch unabhängigen Quellsignalen und einer Überlagerung nach

$$\mathbf{X}(f) = \mathbf{H}(f) \cdot \mathbf{S}(f) \tag{2.66}$$

mit den Mikrofonsignalen $\mathbf{X}(f)$ und der Quellsignale $\mathbf{S}(f)$ im Frequenzbereich vorausgesetzt. Die Übertragungsfunktion $\mathbf{H}(f) \in \mathbb{C}^{M \times Q}$ beschreibt vollständig die Raumeigenschaften zwischen jeder der Q Quellen und jedem der M Mikrofone je Frequenzstützstelle. Hieraus definiert man den Zustand der State Coherence Transform (SCT) im Frequenzbereich als

$$c_{ij,q}(f) = \frac{|H_{iq}(f)|}{|H_{jq}(f)|} \, \mathrm{e}^{\,-\mathrm{j}2\pi f \tau_{ij,q}} \approx \mathrm{e}^{\,-\mathrm{j}2\pi f \tau_{ij,q}} \,. \tag{2.67}$$

Bei nahezu gleichem Abstand der Quelle zu den zwei Mikrofonen i und j können die betragsmäßigen Dämpfungsanteile $|H_{iq}(f)|$ und $|H_{jq}(f)|$ als vernachlässigbar angesehen werden [Nesta u. Omologo, 2012b]. Die positionsabhängige SCT erhält man nach (2.4) als

$$c_{ij,q}(f, \underline{p}_q) = \mathrm{e}^{\,-\mathrm{j}2\pi f \frac{d_{ij}(\underline{p}_q)}{c_0}} \,. \tag{2.68}$$

Unter der Annahme $M = Q$ und dass $\mathbf{H}$ nichtsingulär ist, sucht BSS mittels ICA die inverse Systemmatrix $\mathbf{W} = \mathbf{H}^{-1}$, so dass die Quellsignale nach

$$\tilde{\mathbf{S}}(f) = \mathbf{W}(f)\mathbf{X}(f) = \mathbf{H}^{-1}(f)\mathbf{H}(f)\mathbf{S}(f) = \mathbf{S}(f) \tag{2.69}$$

zurückgewonnen werden können. Dabei besteht für BSS das Problem der Zuordnung der unterschiedlichen Frequenzanteile zu den einzelnen Quellen – auch Permutationsproblem genannt – und eine mögliche Skalierung der Quellen aufgrund von unterschiedlichen Intensitäten der Quellsignale. Es ergibt sich die allgemeinere Formulierung für

$\mathbf{W}(f)$ mit der Permutationsmatrix $\mathbf{\Pi}$, der diagonalen Skalierungsmatrix $\mathbf{D}$ und der zufälligen $(Q \times M)$-dimensionalen Rauschmatrix $\mathbf{R}$

$$\mathbf{W}(f) \approx \mathbf{D}(f)\mathbf{\Pi}(f)\mathbf{H}^{-1}(f) + \mathbf{R}. \tag{2.70}$$

Unter der Voraussetzung, dass das Permutationsproblem gelöst ist, $\mathbf{\Pi} = \mathbf{I}$, und unter Vernachlässigung der Skalierung bei nahezu gleichen Quellen-Mikrofonabständen, kann die inverse Systemmatrix bestimmt und die Signale getrennt werden.

Die geschätzte Systemmatrix berechnet sich dann aus $\hat{\mathbf{H}}(f) = \mathbf{W}^{-1}(f)$ und es ergeben sich die geschätzten SCT-Zustände

$$\hat{c}_{ij,q}(f) = \frac{\tilde{c}_{ij,q}(f)}{|\tilde{c}_{ij,q}(f)|}, \quad \tilde{c}_{ij,q}(f) = \frac{\hat{H}_{iq}(f)}{\hat{H}_{jq}(f)} \tag{2.71}$$

und die Kostenfunktion des ICA-SCT

$$\begin{aligned} P_{\mathrm{ICA-SCT}} &= \sum_{q=1}^{Q} \int_{-\infty}^{\infty} \sum_{1 \le i < j \le M} \sqrt{\left(\hat{c}_{ij,q}(f) - c_{ij,q}(f, \underline{p}_q)\right)^2} \mathrm{d}f \\ &= \sum_{q=1}^{Q} \int_{-\infty}^{\infty} ||\underline{\hat{c}}_q(f) - \underline{c}_q(f, \underline{p}_q)|| \mathrm{d}f. \end{aligned} \tag{2.72}$$

Die Minimierung von $P_{\mathrm{ICA-SCT}}$ liefert schlussendlich die Q Quellpositionen $\underline{p}_q$. Trotz der vielen Möglichkeiten, die ICA-SCT bietet wie die Schätzung der Raumimpulsantworten, ist dieses Verfahren aus folgenden Gründen nur bedingt für diese Arbeit geeignet:

- Aufgrund der mehrdimensionalen, nichtlinearen Optimierung von (2.72) ist ICA-SCT sehr rechenaufwändig und nicht für die Echtzeit-Lokalisierung geeignet.
- Das Lösen des Permutationsproblems ist nicht trivial und bedingt weiterer Rechenschritte oder einer sequentiellen Lokalisierung der Quellen, siehe [Loesch, 2013].
- ICA-SCT ist auch für den überbestimmten Fall $M > Q$ anwendbar, siehe [Loesch, 2013]. Für den unterbestimmten Fall $M < Q$ können maximal die M dominantesten Quellen gefunden werden. Somit gilt die Einschränkung $Q \le M$, die für andere Lokalisierungsverfahren nicht notwendig ist.

2.3.5 Lokalisierung über Beamforming

Im Gegensatz zu dem frequenzselektiven Ansatz mittels ICA stellt der Steered Response Power (SRP) Ansatz ein weniger beschränktes Verfahren dar. Da er der Klasse der Beamformer angehört, hängt er nur von den Verzögerungen entlang der Direktpfade ab

und nicht von der Invertierbarkeit der Systemmatrix und der Lösung des Permutationsproblems. Zudem werden in stark hallenden Umgebungen die Rausch- und Sprachanteile stark korreliert sein, was die Trennbarkeit bei ICA-Verfahren beeinträchtigt. Bei dem SRP-Ansatz wird abhängig von der Quellposition die Summe der GCC-Werte nach (2.55) von allen Mikrofonpaaren gebildet

$$P_{\mathrm{SRP-PHAT}} = \sum_{1 \leq i < j \leq M} |r_{ij}(\tau_{ij}(\underline{p}_q))| \tag{2.73}$$

$$\text{mit} \quad \tau_{ij}(\underline{p}_q) = \frac{d_{ij}(\underline{p}_q)}{c_0}. \tag{2.74}$$

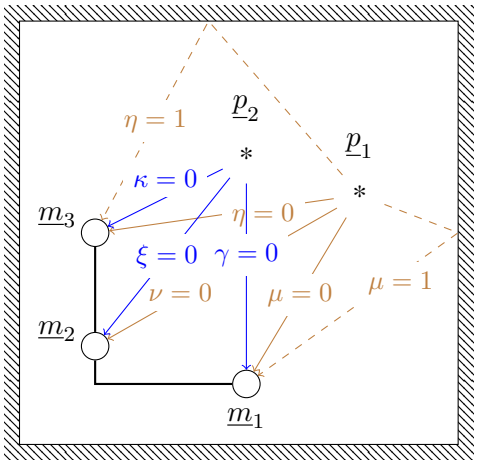
Es hat sich gezeigt, dass auch hier der PHAT-Vorfilter in der GCC r_{ij} die besten Ergebnisse liefert [Dibiase u. a., 2001]. Da $P_{\mathrm{SRP-PHAT}}$ eine nichtlineare Funktion ist, ergeben sich mehrere Maxima, welche im Mehrquellenfall weiter zunehmen. Daher wird $P_{\mathrm{SRP-PHAT}}$ an diskreten Stützstellen im Raum ausgewertet. Anhand des Abstands der Stützstellen kann die Rechendauer und die Genauigkeit von SRP-PHAT beeinflußt werden. Weitere Verbesserungen bzgl. der Rechenzeit ergeben sich aus der *stochastic region contraction*, welche zunächst ein grobes Raster anwendet und anschließend an den Positionen der Maxima eine verfeinerte Suche startet [Do u. a., 2007].

In [Loesch, 2013] wurde gezeigt, dass SRP-PHAT im Gegensatz zu ICA-SCT auch mit kurzen Datenblöcken gute Ergebnisse erzielen kann, was einer schnellen Verarbeitung zuträglich ist.

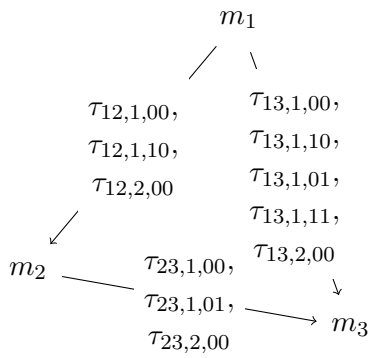
Kapitel 3

Konsistente Graphen

Um die in Kap. 2.3.1 beschriebenen Mehrdeutigkeiten in der Zuordnung der TDOA zu den jeweiligen Quellen zu reduzieren oder sogar aufzulösen, führten [Scheuing u. Yang, 2008] eine graphentheoretische Beschreibung des Problems ein. Bei dieser abstrakten Beschreibung werden die Mikrofone als Knoten und die TDOA als Gewichte auf den Kanten zwischen den Knoten repräsentiert wie in Abb. 3.1 gezeigt. Hier ist in Abb. 3.1(a) ein Lokalisierungsszenario mit zwei Quellen und Mehrwegeausbreitung gezeigt, welches in Abb. 3.1(b) zu einem Graphen mit drei Mikrofonen (Knoten) und den zugehörigen TDOA $\tau_{ij,q,\mu\nu}$ (Kantengewichte) zum Mikrofonpaar ij, der Quelle q und den Ausbreitungswegen μ und ν nach (2.58) abstrahiert ist.



(a) Lokalisierungsbeispiel mit drei Mikrofonen und zwei Quellen in reflektionsbehafteter Umgebung



(b) Abstrahierter Graph mit drei Knoten und den entstehenden TDOA als Kantengewichte

Abbildung 3.1: Abstraktion des Mehrquellen-Lokalisierungsproblems zu einem Graphen

Ziel ist es, die einer Quelle zugehörigen Direktpfad-TDOA bzw. Kantengewichte zu finden. Dabei nutzen wir die notwendige Bedingung (2.65) aus, dass die Summe von

TDOA-Werten entlang einer geschlossenen Masche von Mikrofonen bzw. Knoten Null ergibt, wenn die Werte von der gleichen Quelle und dem gleichen Ausbreitungspfad stammen. Eine solche TDOA-Kombination wird konsistent genannt.

In dieser Arbeit wird ein Verfahren für die Synthese konsistenter Graphen eingeführt, weshalb in diesem Kapitel die graphentheoretischen Grundlagen zusammengefasst werden. Neben der Notation und bereits bekannten Algorithmen aus der Graphentheorie wird auch die Analyse eines Graphen bezüglich seiner Konsistenz diskutiert.

3.1 Notationen und Definition der Konsistenz

Ein Graph $G = (V, E)$ ist definiert als das geordnete Paar bestehend aus der Knotenmenge $V(G) = \{v_1, \ldots, v_M\}$ und der Kantenmenge $E(G) = \{e_n = \{v_i, v_j\} : 1 \leq i < j \leq M, 1 \leq n \leq N\}$. Im Folgenden werden die beiden Mengen als V und E bezeichnet. Ein ungerichteter Graph besitzt die ungeordneten Paare $\{v_i, v_j\}$ als Kanten und ein gerichteter Graph die geordneten Paare (v_i, v_j) mit dem Startknoten v_i und dem Endknoten v_j. Zwei Knoten v_i und v_j sind *adjazent*, wenn die zugehörige Kante in E existiert. Dann werden sie im Folgenden auch als Nachbarn bezeichnet. Die Adjazenzmatrix $\mathbf{A} = [a_{ij}]$ mit

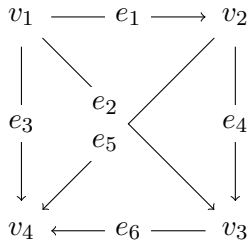
$$a_{ij} := \begin{cases} 1 & \text{falls } \{v_i, v_j\} \in E \\ 0 & \text{sonst} \end{cases}$$

hat die Dimension $M \times M$ und ist für einen ungerichteten Graphen symmetrisch. Eine Erweiterung der Adjazenzmatrix sind die Adjazenzlisten A, die für einen Knoten nur die benachbarten Knoten enthalten, z.B. $A = (\{v_2\}, \{v_1, v_3\}, \ldots)$, wenn der Knoten v_1 mit v_2 und der Knoten v_2 mit v_1 und v_3 verbunden ist. Die Adjazenzlisten liefern eine effiziente Implementierungsmöglichkeit, weil nur Kanten gespeichert werden, die vorhanden sind im Vergleich zur Adjazenzmatrix, die immer $M \times M$ Speicherelemente benötigt.

Ein Knoten v_i und eine Kante e_n sind *inzident*, wenn $e_n = \{v_i, v_j\} \in E$ gilt. Daraus ergibt sich eine weitere Darstellung besonders gerichteter Graphen in der Inzidenzmatrix $\mathbf{Z}$. Die Zeilen repräsentieren die Kanten und die Spalten die Knoten, wobei ein Startknoten einer Kante mit 1 und ein Endknoten mit -1 gekennzeichnet wird. Für den

gerichteten Graphen aus Abb. 3.2 ist die zugehörige Inzidenzmatrix

$$\mathbf{Z} = \begin{bmatrix} & v_1 & v_2 & v_3 & v_4 \\ e_1 & 1 & -1 & 0 & 0 \\ e_2 & 1 & 0 & -1 & 0 \\ e_3 & 1 & 0 & 0 & -1 \\ e_4 & 0 & 1 & -1 & 0 \\ e_5 & 0 & 1 & 0 & -1 \\ e_6 & 0 & 0 & 1 & -1 \end{bmatrix} \tag{3.1}$$



$$\underline{w} = \begin{bmatrix} 3 \\ 1 \\ 4 \\ -2 \\ 1 \\ 3 \end{bmatrix}$$

Abbildung 3.2: Ein gerichteter Graph mit $M = 4$ Knoten und $N = 6$ Kanten und eine konsistente Kantengewichtsbelegung $\underline{w}$.

Der *Grad* eines Knotens ist die Anzahl seiner inzidenten Kanten. Für gerichtete Graphen unterscheidet man zudem noch zwischen dem Eingangs- und Ausgangsgrad, was der Anzahl der adjazenten Start- und Endknoten entspricht.

Sind die Endknoten einer Kante ein und derselbe Knoten, wird die Kante als Schleife bezeichnet. Existieren parallele Kanten zwischen zwei Knoten, so wird der Graph Multigraph genannt. In dieser Arbeit werden nur *einfache Graphen* betrachtet, die keine parallelen Kanten und keine Schleifen haben [Jungnickel, 2008; Diestel, 2006; Gutin, 2008].

Als Pfad $P = v_1 e_i v_2 e_j v_3 \ldots e_l v_k$, mit $e_i = \{v_1, v_2\}$, $e_j = \{v_2, v_3\}$ und $e_l = \{v_m, v_k\}$, wird eine alternierende Sequenz von inzidenten Knoten und Kanten bezeichnet. Besitzen zwei Pfade P_1 und P_2 keine gemeinsame Kante, so nennt man sie *kantendisjunkt* und wenn sie keinen gemeinsamen Knoten haben *knotendisjunkt*. Knotendisjunkte Pfade sind immer kantendisjunkt, wobei die Umkehrung nicht gilt.

Existiert zwischen allen N Knotenpaaren in einem Graphen ein Pfad, so heißt der Graph *zusammenhängend*. Sind alle Knotenpaare adjazent, ist der Graph *vollständig zusammenhängend*. In diesem Fall gilt für den einfachen Graphen, dass er eine maximale Anzahl von $N_{\max} = \binom{M}{2}$ Kanten hat, wie es in Abb. 3.2 der Fall ist.

Fallen die Endknoten eines Pfades auf den gleichen Knoten, so ist der Pfad geschlossen und wird *Zyklus* oder *Masche* genannt. Besitzt ein Graph nur Pfade und keine einzige

Masche, so wird er *azyklisch* genannt. Eine Masche der Länge k beinhaltet genau k Knoten und k Kanten. Ein Graph ist *gewichtet*, wenn ein Kantengewicht $w_n : e_n \to \mathbb{R}$ für jede Kante $e_n \in E$ existiert. Demnach ist eine Kantengewichtsbelegung als $\underline{w} = [w_1, w_2, \ldots, w_N]^T$ definiert. Zusammen mit dem entsprechenden Graphen G, der die zugrundeliegende Topologie festlegt, ist ein gewichteter Graph gegeben als das geordnete Paar $G^w = (G, \underline{w})$.

Definition 1. *Ein gerichteter, gewichteter Graph $G^w = (G, \underline{w})$ heißt konsistent, wenn er entlang aller Maschen eine Nullsumme von Kantengewichten aufweist.*

Der Graph G^w in Abb. 3.2 ist konsistent. Eine Besonderheit in der Signalverarbeitung zur Graphentheorie ist die Invertierung der Kantenrichtung. Daraus folgt eine Negierung des jeweiligen Kantengewichtes. Für eine Kante $e_n = (v_i, v_j)$ und das zugehörige Gewicht w_{ij} bedeutet das konkret $w_{ij} = -w_{ji}$. Werden die Kantengewichte w_n aus einer Kantengewichtsmenge $W_n = \{w_{n,1}, \ldots, w_{n,K_n}\}$ ausgewählt, spannt $W = W_1 \times \cdots \times W_N$ den Raum der möglichen Kantengewichtskombinationen auf.

3.2 Maschen und Graphoid

Um die Konsistenz eines Graphen G^w zu überprüfen, müssen seine Maschen gefunden und analysiert werden. Zu dem Beispiel aus Abb. 3.2 sind alle Maschen in Abb. 3.3 aufgelistet.

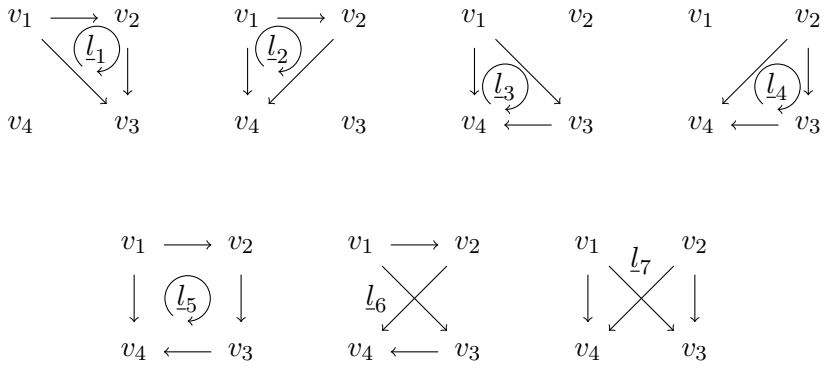


Abbildung 3.3: Alle Maschen für den Graph aus Abb. 3.2

Diese werden in der *Maschenmatrix* $\mathbf{L}$ gespeichert, welche für die Maschen aus Abb.

3.3 wie folgt aussieht

$$\mathbf{L} = \begin{bmatrix} & \underline{l}_1 & \underline{l}_2 & \underline{l}_3 & \underline{l}_4 & \underline{l}_5 & \underline{l}_6 & \underline{l}_7 \\ e_1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ e_2 & -1 & 0 & 1 & 0 & 0 & -1 & 1 \\ e_3 & 0 & -1 & -1 & 0 & -1 & 0 & -1 \\ e_4 & 1 & 0 & 0 & 1 & 1 & 0 & -1 \\ e_5 & 0 & 1 & 0 & -1 & 0 & 1 & 1 \\ e_6 & 0 & 0 & 1 & 1 & 1 & -1 & 0 \end{bmatrix}. \tag{3.2}$$

Jede Spalte entspricht einer Masche $\underline{l} \in \{-1, 0, 1\}^N$, die eine 0 hat, wenn die zugehörige Kante nicht beinhaltet ist, 1 wenn die zugehörige Kante im Sinne der Umlaufrichtung der Masche durchlaufen wird und -1 falls sie entgegengesetzt durchlaufen wird.

3.2.1 Anzahl aller Maschen

Ein einfacher Graph mit M Knoten kann maximal $N_{\max} = \frac{M(M-1)}{2}$ Kanten haben. Für alle Maschen der Länge m lassen sich $m!$ unterschiedliche Reihenfolgen der Knoten finden. Da die zyklischen Verschiebungen keinen Unterschied machen, sind $v_1, v_2, \ldots, v_m$ und die um i verschobenen Varianten $v_i, v_{i+1}, \ldots, v_m, v_1, \ldots, v_{i-1}$, von denen es m Unterschiedliche gibt, als gleich zu betrachten. Außerdem sind umgekehrte Durchlaufrichtungen $v_1, \ldots, v_{m-1}, v_m$ und $v_m, v_{m-1}, \ldots, v_1$ ebenfalls als gleichbedeutend anzusehen. Daraus lässt sich die Gesamtanzahl N_{L} der Maschen in einem vollständigen Graphen zu

$$N_{\mathrm{L}} = \sum_{m=3}^{M} \binom{M}{m} \frac{m!}{m \cdot 2} = \sum_{m=3}^{M} \binom{M}{m} \frac{(m-1)!}{2} \tag{3.3}$$

bestimmen. Da

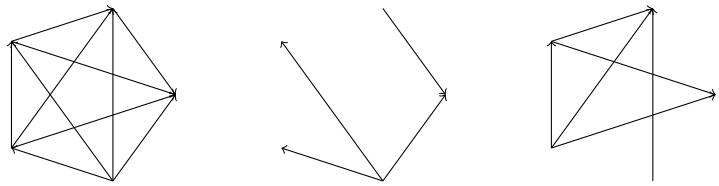
$$N_{\mathrm{L}} = \frac{1}{2} \sum_{m=3}^{M} \frac{M!}{m \cdot (M-m)!} = \frac{1}{2} \sum_{m=3}^{M} \frac{1}{m} \prod_{n=m+1}^{M} n > \frac{1}{2} \sum_{m=3}^{M} m^{M-m-1} \tag{3.4}$$

gilt, ist die Anzahl aller Maschen N_{L} mit der Knotenanzahl M exponentiell steigend. Deshalb ist es nicht effizient, alle Maschen in einem Graphen auf Konsistenz zu überprüfen.

Bekannte Verfahren zur Bestimmung aller Maschen sind von [Tarjan u. Read, 1975] und [Johnson, 1975] gegeben. Diese Verfahren sind nur für ungerichtete Graphen angegeben, in denen man die Kanten in beide Richtungen durchlaufen kann. Wegen der Invertierbarkeit der Kantenrichtungen $w_{ij} = -w_{ji}$ können sie in unserem Fall auch auf den gerichteten Graphen angewandt werden.

3.2.2 Fundamentale Maschen

Um nicht alle Maschen berechnen zu müssen, sucht man einen minimalen Satz von Maschen, der ausreicht, um die Konsistenz zu überprüfen. In [Balabanian u. Bickart, 1969] wird hierzu ein Verfahren gezeigt, das für die Maschenanalyse in elektrischen Netzwerken verwendet wird. Zuerst wird ein aufspannender Baum G_{AB} berechnet, der ein azyklischer Teilgraph von G ist. Für den aufspannenden Baum gilt dabei, dass alle Knoten erreicht werden, aber keine Masche geschlossen wird. Ein möglicher aufspannender Baum für den Graph aus Abb. 3.4(a) ist in Abb. 3.4(b) gegeben.



(a) Graph (b) Aufspannender Baum (c) Kobaum

Abbildung 3.4: Ein gerichteter Graph mit $M = 5$ Knoten, sowie ein möglicher aufspannender Baum und der zugehörige Kobaum

Es gilt allgemein, dass $V(G_{\mathrm{AB}}) = V(G)$ und $E(G_{\mathrm{AB}}) \subseteq E(G)$ mit $|E(G_{\mathrm{AB}})| = M - 1$ ist. Die restlichen $N - M + 1$ Kanten bilden den sogenannten komplementären Baum G_{KB}. Für den komplementären Baum, auch Kobaum genannt, gilt $V(G_{\mathrm{KB}}) \subseteq V(G)$ und $E(G_{\mathrm{KB}}) = E(G) \setminus E(G_{\mathrm{AB}})$. Für den aufspannenden Baum aus Abb. 3.4(b) ist der Kobaum in Abb. 3.4(c) gegeben.

Um den aufspannenden Baum zu berechnen, werden unterschiedliche Verfahren angewendet. Die bekanntesten Vertreter sind hierbei die Breitensuche (Breadth-First-Search, BFS) und die Tiefensuche (Depth-First-Search, DFS), welche im Folgenden genauer vorgestellt werden.

Breitensuche

Um das Problem des Handelsreisenden zu lösen, entwickelte Dijkstra die BFS-Suche, um die kürzeste Verbindung aller Knoten in einem Graphen zu finden [Dijkstra, 1959]. Das Vorgehen ist ein sukzessives Abwandern der Knoten, die in der Warteschlange Q gespeichert sind. Diese arbeitet nach einem first-in-first-out (FIFO) Prinzip, sodass Knoten, die zuerst darin gespeichert werden, auch zuerst besucht werden. An jedem besuchten Knoten sucht man nach allen adjazenten Knoten und speichert die inzidenten Kanten im aufspannenden Baum T. Danach löscht man die adjazenten Knoten aus der Knotenmenge V und speichert diese in der Warteschlange Q. Anschließend

Algorithmus 1: Pseudocode der Breitensuche

Input : BFS(V, E, n), Knotenmenge V, Kantenmenge E, Startknoten n
Output : Kantenmenge T, besuchte Knoten $\underline{b}$

```
T = ∅, b = 0_M, Q = (n);
while Q ≠ ∅ do                          // solange noch Elemente in der FIFO Schlange
    n = q_1 ∈ Q ;                       // nehme erstes Element aus Q
    Q = Q \ q_1 ;                       // lösche dieses Element aus Q
    for m ∈ V do
        if {n, m} ∈ E ∧ b(m) ≠ 1 then   // suche nach nicht besuchten adjazenten Knoten
            b(m) = 1 ;                  // markiere ihn als besucht
            T = T ∪ {n, m};             // hänge Kante an den Baum
            V = V \ m ;                 // entferne Knoten aus der weiteren Suche
            Q = (Q, m) ;                // hänge neuen Knoten an das Ende von Q
        end
    end
end
```

wird der nächste Knoten aus der Warteschlange entnommen und verarbeitet, bis die Warteschlange leer ist. Der BFS-Algorithmus ist in Alg. 1 dargestellt und wird mit BFS($V(G) \setminus v_0, E(G), v_0$) aufgerufen. Dabei ist v_0 der Start- oder Wurzelknoten der Suche.

Die BFS-Suche führt zu einem weit aufspannenden Baum, da zuerst alle inzidenten Kanten des Wurzelknotens in den Baum aufgenommen werden und erst danach, falls nicht besuchte Knoten existieren, weitere Knoten besucht und abgearbeitet werden.

Das BFS-Verfahren ist in polynomieller Zeit mit $\mathcal{O}(|V| + |E|)$ Rechenoperationen ein effizientes Verfahren und deshalb gut geeignet für die Suche nach dem aufspannenden Baum [Russell u. Norvig, 2010].

Tiefensuche

Eine gänzlich andere Strategie stellt die DFS-Suche dar. Sobald ein zum aktuellen Knoten n adjazenter Knoten m gefunden wurde, wird eine neue Suche von diesem Knoten m gestartet. Dabei werden die inzidente Kante $\{n, m\}$ aus der Kantenmenge E gelöscht und an den aufspannenden Baum T angehängt. Der aktuelle Knoten n wird für die nachfolgenden Rekursionen entfernt und der adjazente Knoten m als besucht markiert. Sind in einem Rekursionsaufruf bereits alle Knoten besucht worden, so kehrt DFS eine Rekursion zurück und sucht nach weiteren unbesuchten Nachbarn. Sollten keine mehr vorhanden sein, so geht DFS Schritt für Schritt zurück und sucht somit alle zusammenhängenden Knoten ab. Der rekursive DFS-Algorithmus ist in Alg. 2 dargestellt. Der erste Aufruf geschieht mit dem Wurzelknoten v_0 durch DFS($V(G) \setminus v_0$, $E(G)$, v_0).

Algorithmus 2: Pseudocode der Tiefensuche

Input : DFS(V, E, n), Knotenmenge V, Kantenmenge E, Startknoten n
Output : Kantenmenge T, besuchte Knoten $\underline{b}$
$T = \emptyset$, $\underline{b} = \underline{0}_M$;
recDFS(V, E, n, T, $\underline{b}$);

Funktion recDFS(V, E, n, T, $\underline{b}$)
for $m \in V$ **do**
  if $\{n, m\} \in E \wedge \underline{b}(m) \neq 1$ **then** *// falls der Knoten m adjazent und unbesucht ist*
    $\underline{b}(m) = 1$; *// markiere ihn als besucht*
    $T = T \cup \{n, m\}$; *// füge die inzidente Kante an den aufspannenden Baum*
    $E = E \setminus \{n, m\}$; *// entferne die inzidente Kante und den adjazenten Knoten*
    $V = V \setminus m$;
    recDFS(V, E, m, T, $\underline{b}$); *// starte eine erneute Suche vom adjazenten Knoten*
  end
end

Die DFS-Suche wandert im Vergleich zu BFS vom Ursprungsknoten weg und erzeugt somit einen schmalen und tief aufspannenden Baum. Dabei benötigt der Algorithmus die gleiche Komplexität von maximal $\mathcal{O}(|V| + |E|)$ Operationen [Russell u. Norvig, 2010].

Kobaum und fundamentale Maschen

Für den Graphen in Abb. 3.5 a) sind beispielhaft ein aufspannender Baum aus einer BFS-Suche in b) und einer DFS-Suche in c) gezeigt.

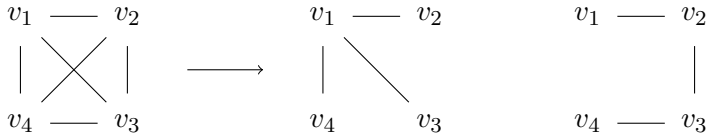


Abbildung 3.5: Gegeben der Graph in a) und der Wurzelknoten v_1, so entsteht mit einer BFS-Suche der aufspannende Baum in b) und mit DFS der Baum in c)

Um nun die minimale Anzahl an fundamentalen Maschen (FM) zu bekommen, werden die restlichen Kanten des Kobaums $E(G_{\mathrm{KB}})$ nacheinander an den aufspannenden Baum gefügt. Damit wird jeweils eine FM geschlossen. In Abb. 3.6 ist dies für die aufspannenden Bäume aus Abb. 3.5 b) und c) gezeigt.

Aus dem aufspannenden Baum und dem daraus resultierenden Kobaum können die FM mittels einer simplen DFS-Suche geschlossen werden. Der Algorithmus ist in Alg. 3

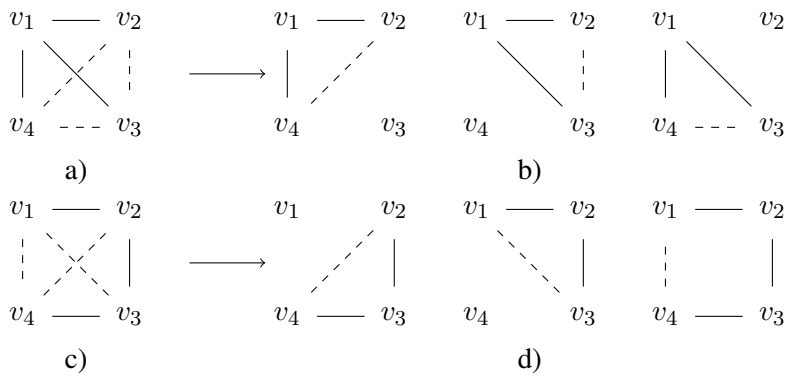


Abbildung 3.6: Aufspannender Baum (durchgezogen) und Kobaum (gestrichelt) von Abb. 3.5 b) in a) und von Abb. 3.5 c) in c). Die FM mit je einer Kante des Kobaums sind in b) bzw. d) dargestellt.

dargestellt und wird mit CLOSEFM($v_{\text{start}}(e_{\text{KB}})$, $v_{\text{end}}(e_{\text{KB}})$, $E(G_{\text{AB}})$, $\underline{l}_f$) gestartet. Hierbei ist $\underline{l}_f \in \{-1, 0, 1\}^N$ der Maschenvektor der FM gemäß der Definition aus (3.2), der zunächst mit Nullen initialisiert wird $\underline{l}_f = \underline{0}_N$. Mit e_{KB} ist die aktuelle Kobaumkante referenziert, die in $\underline{l}_f$ als 1 geschrieben wird, da sie die Umlaufrichtung der FM festlegt.

Daraus folgt, dass die minimale Anzahl der FM genau der Anzahl der Kobaumkanten entspricht, $N_{\text{FM}} = N - M + 1$. Diese Anzahl wächst maximal quadratisch mit der Anzahl der Knoten, im Vergleich zum exponentiellen Anstieg aller Maschen N_{L} aus (3.3), wie in Abb. 3.7 veranschaulicht.

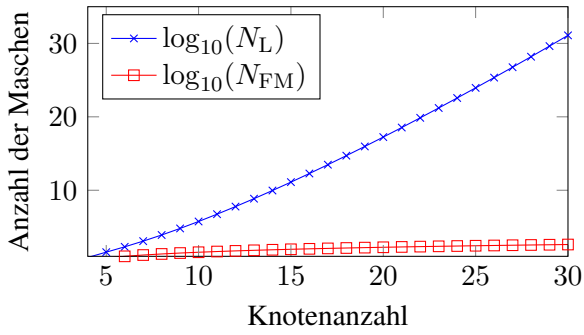


Abbildung 3.7: Mit der Knotenanzahl wächst die maximale Anzahl der Maschen exponentiell und die Anzahl der FM quadratisch

Algorithmus 3: Pseudocode um eine Masche zu schließen

Input : CLOSEFM(v_{start}, v_{end}, T, $\underline{l}_f$), Startknoten v_{start}, Zielknoten v_{end}, aufspannender Baum T, FM $\underline{l}_f$
Output : FM $\underline{l}_f$, boolesche Variable *closed*

```
for e ∈ T do
    if v_start ∈ e then
        T = T \ e;
        if v_start = v_start(e) then
            l_f(e) = 1;;                    // Kanten- und Maschenrichtung sind gleich
            v_next = v_end(e);
        else
            l_f(e) = −1; ;                  // Kanten- und Maschenrichtung sind entgegengesetzt
            v_next = v_start(e);
        end
        if v_next = v_end then              // ist die Masche geschlossen
            closed = 1;
        else
            closed = 0;
            CLOSEFM(v_next, v_end, T, l_f);
            if NOT(closed) then             // Masche konnte nicht geschlossen werden
                l_f(e) = 0;                 // Zurücksetzen der Kante in der Masche
            end
        end
    end
end
```

Kombination von Maschen

Gegeben zwei beliebige Maschen $\underline{l}_i$ und $\underline{l}_j$, so sind ihre äquivalenten Kantenmengen $C_i = \{e_n \in E : \underline{l}_i(n) \neq 0\}$ und $C_j = \{e_n \in E : \underline{l}_j(n) \neq 0\}$. Die Kombination der Maschen ist die symmetrische Differenz $C_i \Delta C_j = C_i \cup C_j - C_i \cap C_j$ [Gutin, 2008]. Graphisch entfallen somit bei der Kombination gemeinsame Kanten und nur solche Kanten, die in nur einer Masche vorkommen, werden beibehalten.

In Abb. 3.8 sind alle Kombinationen der FM aus Abb. 3.6 b) gezeigt. Es ist leicht zu sehen, dass die Maschen in Abb. 3.8 genau allen Maschen aus Abb. 3.3 entsprechen. Dieser Zusammenhang werden wir im Folgenden genauer betrachten.

Fundamentale Maschenmatrix zur Konsistenzprüfung

Die FM $\underline{l}_f$ werden in der fundamentalen Maschenmatrix $\mathbf{L}_{\mathrm{FM}} = [\underline{l}_{f,i}]$ $(1 \leq i \leq N_{\mathrm{FM}})$ abgespeichert. Da $\mathbf{L}_{\mathrm{FM}}$ jede Kante des Kobaums genau einmal beinhaltet, kann sie auch

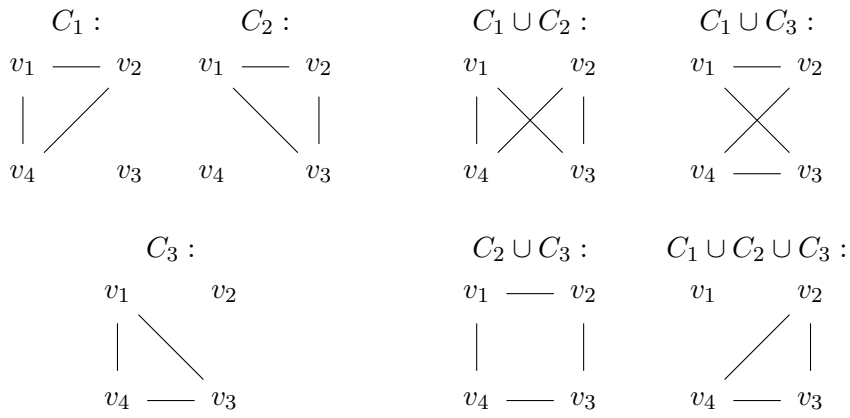


Abbildung 3.8: Alle Kombinationen der FM aus Abb. 3.6 b)

in der Form

$$\mathbf{L}_{\mathrm{FM}} = \begin{bmatrix} \mathbf{L}_{\mathrm{AB}} \\ \mathbf{L}_{\mathrm{KB}} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{\mathrm{AB}} \\ \mathbf{I}_{(N-M-1)} \end{bmatrix} \tag{3.5}$$

geschrieben werden. Der Rang von $\mathbf{L}_{\mathrm{FM}}$ ist somit $\mathrm{Rang}(\mathbf{L}_{\mathrm{FM}}) = N - M + 1$ [Balabanian u. Bickart, 1969], d. h. die Spalten von $\mathbf{L}_{\mathrm{FM}}$ spannen einen $(N - M + 1)$-dimensionalen Unterraum des $\{-1, 0, 1\}^N$ auf.

Der Spaltenraum der Maschenmatrix $R(\mathbf{L})$ in (3.2) sowie der Spaltenraum der FM-Maschenmatrix $R(\mathbf{L}_{\mathrm{FM}})$ liegen im Kantenraum $\mathcal{E}(G) = \mathbb{F}_2^N$. Dabei ist jede Kante eine Abbildung auf das Galois-Feld $e \rightarrow \{0, 1\} = \mathbb{F}_2$ [Diestel, 2006; Gutin, 2008; Tutte, 2001].

Satz 1. *Der Maschenraum $R(\mathbf{L})$ ist ein $(N - M + 1)$-dimensionaler Unterraum des $\mathbb{F}_2^N$ und die FM stellen eine linear unabhängige Basis dieses Vektorraums dar.*

Beweis. Für alle $M-1$ Kanten des aufspannenden Baumes gilt, dass sie zunächst in keiner Masche enthalten ist. Somit kann der Rang der Maschenmatrix maximal $\mathrm{Rang}(\mathbf{L}) \leq N-M+1$ sein. Da der aufspannende Baum der maximale, azyklische Teilgraph ist, gilt die Gleichheit und somit ist der Maschenraum $R(\mathbf{L})$ ein $(N - M + 1)$-dimensionaler Unterraum des $\mathbb{F}_2^N$.

Des Weiteren gilt für jede FM im Speziellen, dass sie aus einer Kante des Kobaums erzeugt wird und deshalb ist jede FM linear unabhängig von der anderen. Darum ist die FM-Matrix mit ihren $N - M + 1$ Spaltenvektoren eine Basis des Maschenraums $R(\mathbf{L})$. □

Demnach sind maximal $N - M + 1$ Spaltenvektoren in (3.2) linear unabhängig und es genügt für die Konsistenz nur die linear unabhängige Basis, die die Maschen von $\mathbf{L}_{\mathrm{FM}}$

darstellen, zu prüfen. Für einen gewichteten Graphen G^w mit der Kantengewichtsbelegung $\underline{w}$ ist die Konsistenzbedingung mit

$$\mathbf{L}_{\mathrm{FM}}^T \underline{w} = \underline{0} \tag{3.6}$$

gegeben.

Schnittraum und Graphoid

Der Spaltenraum der Inzidenzmatrix $\mathbf{Z}$ auf S. 47 ist ebenfalls im Kantenraum $\mathcal{E}(G)$ definiert. Für einen zusammenhängenden Graphen erreichen genau die $M-1$ Kanten des aufspannenden Baumes alle Knoten und sind zudem linear unabhängig, da jede Inzidenzmatrix eines aufspannenden Baumes sich auf die Form

$$\mathbf{Z}_{\mathrm{AB}} = \begin{bmatrix} \underline{1}_{M-1} & -\mathbf{I}_{M-1} \end{bmatrix} \tag{3.7}$$

bringen lässt. Jede weitere Kante bzw. Zeile, die angefügt wird, erhöht den $\mathrm{Rang}(\mathbf{Z}) = M-1$ nicht.

Zwei Vektoren $\underline{e}_i = [\alpha_1, \ldots, \alpha_N]^T$ und $\underline{e}_j = [\beta_1, \ldots, \beta_N]^T$ des Kantenraums $\mathcal{E}(G)$ mit $\alpha_i, \beta_j \in \{-1, 0, 1\}$ sind orthogonal, wenn

$$\underline{e}_i^T \, \underline{e}_j = 0 \tag{3.8}$$

mit $\underline{e}_i^T \, \underline{e}_j = (\alpha_1\beta_1 + \cdots + \alpha_N\beta_N) \bmod 2 \in \mathbb{F}_2$ gilt [Gutin, 2008; Tutte, 2001; Jungnickel, 2008]. Somit sind nur solche Vektoren orthogonal, die eine gerade Anzahl von Kanten gemeinsam haben.

Satz 2. *Für alle Spaltenvektoren $\underline{z} \in \mathbf{Z}$ und die Spaltenvektoren aus $\underline{l} \in \mathbf{L}$ ist $\underline{z}^T \underline{l} = 0$.*

Beweis. Zum einen beinhaltet die Inzidenzmatrix eine Relation zwischen den ein- und ausfallenden Kanten und den jeweiligen Knoten. Zum andern stellt die Maschenmatrix eine Maskierung dar, die nur die an der jeweiligen Masche beteiligten Kanten selektiert. Somit kann das Produkt $\underline{z}^T \underline{l}$ als eine Summe der ein- und ausfallenden Kanten entlang einer Masche $\underline{l}$ verstanden werden. Daher löscht sich jede Kante einer Masche, multipliziert mit dem Startknoten, als ausfallende Kante und, multipliziert mit dem Endknoten, als einfallende Kante aus und es gilt $\underline{z}^T \underline{l} = 0$ [Balabanian u. Bickart, 1969; Gutin, 2008]. □

Damit ist jede Spalte aus $\mathbf{Z}$ orthogonal zu jeder Spalte aus $\mathbf{L}$ und es gilt für die zugehörigen Spaltenräume $R(\mathbf{Z})$ und $R(\mathbf{L})$, dass

$$R(\mathbf{Z}) \perp R(\mathbf{L}) \quad \text{mit} \quad \mathrm{Rang}(\mathbf{Z}) + \mathrm{Rang}(\mathbf{L}) = N. \tag{3.9}$$

Der Unterraum $R(\mathbf{Z})$ stellt einen möglichen *Schnittraum* für den Graphen dar. Ein Schnitt ist eine Kantenmenge, deren Entfernen den Graphen in zwei disjunkte Teilgraphen teilt. Jede Spalte der Inzidenzmatrix repräsentiert alle inzidenten Kanten zu einem Knoten v. Werden diese Kanten entfernt, so wird der Graph $G = (V, E)$ in die Teilgraphen $(v, \emptyset)$ und $(V \setminus v, \tilde{E})$ geteilt mit $\tilde{E} = E \setminus \{\{v, w\} : w \in V\}$. Der Schnittraum, der durch $R(\mathbf{Z})$ repräsentiert wird, ist orthogonal zum Maschenraum bzw. $R(\mathbf{L}_{\mathrm{FM}})$.

Die Dualität zweier Vektorräume ist Gegenstand der *Matroidtheorie*, die sich mit der abstrakten Beschreibung von Abhängigkeit beschäftigt, wie z.B. in der linearen Algebra oder in der kombinatorischen Optimierung [Oxley, 2011]. Für den speziellen Fall der Orthogonalität des Maschen- und Schnittraums in der Graphentheorie spricht man auch von einem *Graphoid* [Gutin, 2008; Tutte, 2001; Jungnickel, 2008].

3.3 Komponentensuche

Da die Synthese beliebige Eingangsgraphen erhalten kann, muss man zunächst die Existenz von Maschen überprüfen. Für einfache, zusammenhängende Graphen ist die Bedingung für die Existenz mindestens einer Masche gegeben durch $N > M - 1$. Diese Bedingung ist direkt aus der Tatsache ableitbar, dass ein azyklischer aufspannender Baum maximal $M - 1$ Kanten besitzt und ab M Kanten mindestens eine Masche geschlossen wird. Diese Bedingung ist aber nicht immer ausreichend, wie der Graph aus Abb. 3.9 mit $N > M$ Kanten zeigt.

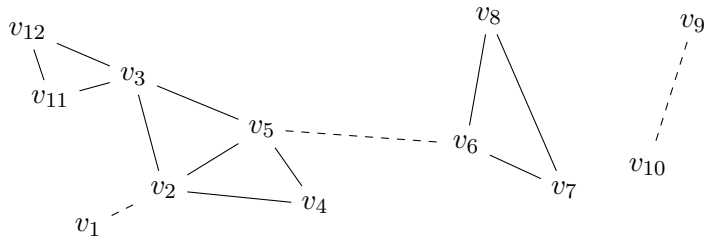


Abbildung 3.9: Unzusammenhängender Graph mit $M = 12$ Knoten und $N = 14$ Kanten. Zweifach zusammenhängende Komponenten sind mit durchgezogenen Kanten gekennzeichnet.

Dieser Graph ist nicht zusammenhängend und die Kanten $\{v_1, v_2\}$, $\{v_5, v_6\}$ und $\{v_9, v_{10}\}$ sind an keiner Masche beteiligt. Bei $\{v_1, v_2\}$ handelt es sich um eine *hängende* Kante, da ein Endknoten nur zu dieser Kante inzident ist. Da diese Kante zu keiner Masche beiträgt, kann sie für die Analyse entfernt werden, wobei die Endknoten erhalten bleiben. Daraus ergibt sich, dass v_1 ein *isolierter* Knoten wird. Isolierte Knoten und isolierte Kanten wie $\{v_9, v_{10}\}$ können für die Konsistenzbedingung ebenfalls entfernt werden. Die Kante $\{v_5, v_6\}$ stellt eine Brücke dar, d. h. einen *Kantenschnitt* der Mächtigkeit

eins und schließt eine kantendisjunkte Masche mit beiden Endknoten aus. Bei v_3 in Abb. 3.9 handelt es sich äquivalent um einen *Schnittknoten* der Mächtigkeit eins. Wenn v_3 entfernt wird – und mit ihm $\{v_{11}, v_3\}, \{v_{12}, v_3\}, \{v_2, v_3\}$ und $\{v_3, v_5\}$ – ist der resultierende Graph nicht zusammenhängend. Die Schnittkanten und Schnittknoten der Mächtigkeit eins haben die Besonderheit, dass alle Knoten in den resultierenden Teilgraphen G_1 und G_2 nur über einen Pfad verbunden sind, der über die Schnittkante oder den Schnittknoten verläuft. Es ist leicht zu erkennen, dass jede kantendisjunkte Masche in G_1 unabhängig von den Maschen in G_2 betrachtet werden kann. Deshalb können alle einfachen Schnittkanten entfernt werden. Für Schnittknoten erfolgt eine Duplizierung des Knoten, wobei jedem resultierenden Teilgraphen eine Kopie mit den jeweiligen inzidenten Kanten zugeordnet wird.

Die Eigenschaft, dass jedes Knotenpaar in einem Graphen über mindestens zwei knotendisjunkte Pfade verbunden ist und somit in der selben Masche liegt, heißt *zweifachzusammenhängend* [Jungnickel, 2008] und ist für die Synthese eine notwendige Bedingung. In [Hopcroft u. Tarjan, 1973] und [Tarjan, 1972] zeigen die Autoren einen Algorithmus, der alle zweifach-zusammenhängenden Teilgraphen, welche im Folgenden *Komponenten* genannt werden, berechnet. Dieser Algorithmus ist in Alg. 4 dargestellt.

Bei Alg. 4 handelt es sich um einen erweiterten DFS-Suchalgorithmus, der zunächst alle zusammenhängenden Knoten u im Sinne ihrer Besuchsreihenfolge durchnummeriert (Zeile 7) und mit ihrem Vorgänger v assoziiert. Der Startknoten ist sein eigener Vorgänger und stellt den ersten Tiefpunkt dar. Der Tiefpunkt ist der Knoten mit der kleinsten Nummer, welcher aktuell erreichbar ist. Der besuchte Knoten u wird auf einem Stapel S gespeichert und als Vorgänger für den nächsten Knoten vermerkt. Wird bei der DFS-Suche ein Knoten u mit bereits zugewiesener Nummer besucht, vergleicht Alg. 4 die Nummer des Knoten u mit dem Tiefpunkt des Vorgängers v und erneuert den Tiefpunkt von v, falls die Nummer des Knoten u niedriger ist (Zeile 13). Der Tiefpunkt ist somit als hierarchisch niedrigster Knoten zu sehen, der über einen anderen Pfad als den aufspannenden Baum erreicht werden kann. Dies gewährleistet den zweifachen Zusammenhang.

Ab dann erfolgt eine Rückwärtsüberprüfung, wobei der Tiefpunkt des Knotens v mit der Nummer des Vorgängers überprüft wird (Zeile 18). Solange der Tiefpunkt von v kleiner ist, wird der Tiefpunkt des Vorgängers angepasst. Ist der Tiefpunkt selber erreicht, wird der besuchte Teilgraph als Komponente K gespeichert, wobei nur die Knoten aus dem Stapel genommen werden, deren Tiefpunkte größer oder gleich dem Tiefpunkt selber sind (ab Zeile 22). Anschließend werden die Knoten der gespeicherten Komponente vom Stapel entfernt, wobei der Schnittknoten selber bleibt.

Ist der letzte Vorgänger des aktuellen Knotens der Startknoten, so erfolgt auch eine Überprüfung, ob es sich um einen Schnittknoten handelt (Zeile 32), und der restliche

Stapel S wird als Komponente gespeichert. Falls der Startknoten ein Schnittknoten ist, muss final überprüft werden, ob noch weitere Knoten vorhanden sind (Zeile 40).

Der Algorithmus ist in Alg. 4 dargestellt und findet nebenbei auch die einfachen Schnittknoten C, die separat ausgegeben werden. Wegen diesen Schnittknoten kann nicht der ganze Stapel S gespeichert werden, sondern nur die Knoten mit höherem Tiefpunkt. Niedrigere Knoten fallen in eine andere Komponente.

Da Alg. 4 von einem zusammenhängenden Graphen mit $N > M - 1$ Kanten ausgeht, wird er alle mit dem Wurzelknoten v_0 zusammenhängenden Komponenten finden. Es muss somit für einen allgemeinen Graphen noch eine Hauptroutine angewendet werden, welche überprüft, ob nach jedem Durchgang noch weitere Kanten vorhanden sind, und Alg. 4 mit einem neuen Wurzelknoten aufruft.

Sind alle Komponenten detektiert, können diese in unabhängige Teilgraphen zerlegt werden und mittels der bekannten Bedingung (3.6) auf Konsistenz überprüft werden. Für den Rest dieser Arbeit ist mit einem Graphen eine Komponente gemeint.

Algorithmus 4: Pseudocode zum Algorithmus von Hopcroft und Tarjan für die Komponentensuche

Input : Ungerichteter Graph $G = (V, E)$, Startknoten v_0
Output : Menge von Komponenten K, Menge der Schnittknoten C

```
1  C = ∅; Initialisiere Adjanzenzliste A; Zähler z = 0; v = v_0;
2  Stapel S = {v}; Nummer(v) = z; Tiefpunkt(v) = z; Vorgänger(v) = −1 ;
3  repeat
4      while A(v) ≠ ∅ do                          // es gibt noch nicht besuchte Nachbarn
5          u ∈ A(v) und A(v) = A(v) \ u;          // nehme einen Nachbarn u von v
6          if Nummer(u) ungültig then
7              Nummer(u) = z; z = z + 1;          // nummeriere u und erhöhe den Zähler
8              Vorgänger(u) = v;                  // setze den Vorgänger auf v
9              Tiefpunkt(u) = v;                  // setze den Tiefpunkt auf v
10             S = S ∪ u;                         // hänge u an den Stapel
11             v = u;
12         else
13             Tiefpunkt(v) = min{Tiefpunkt(v), Nummer(u)};
14         end
15     end
16     if v ≠ v_0 then
17         if Vorgänger(v) ≠ v_0 then             // ist der Vorgänger nicht der Wurzelknoten
18             if Tiefpunkt(v) < Nummer(Vorgänger(v)) then   // ist aktueller Tiefpunkt kleiner
19                 Tiefpunkt(Vorgänger(v)) = min{Tiefpunkt(Vorgänger(v)), Tiefpunkt(v)};
20             else
21                 C = C ∪ Vorgänger(v);          // Vorgänger(v) ist ein Schnittknoten
22                 K = {Vorgänger(v)};            // beginne eine neue Komponente
23                 forall the s ∈ S do
24                     if Tiefpunkt(s) ≥ Tiefpunkt(v) ∧ s ≠ Vorgänger(v) then
25                         K = K ∪ s;             // füge s an die zu speichernde Komponente
26                         S = S \ s;             // entferne s vom Stapel
27                     end
28                 end
29                 Speichere neue zweifach-zusammenhängende Komponente K;
30             end
31         else
32             if A(v) ≠ ∅ then
33                 C = C ∪ v;                     // Wurzelknoten ist ein Schnittknoten
34             end
35             Speichere den Stapel S als neue Komponente K;
36             S = {v_0};                         // initialisiere S erneut
37         end
38         v = Vorgänger(v);
39     end
40 until (Vorgänger(v) == −1) ∧ (A(v) == ∅)   // Wurzelknoten ist erreicht ohne Nachbarn;
```

Kapitel 4

Synthese konsistenter Graphen

In der Synthese konsistenter Graphen wird nicht mehr die Analyse eines gegebenen Graphen $G^w = (G, \underline{w})$ auf Konsistenz betrachtet, sondern vielmehr die folgende Aufgabe:

Problem 1. *Gegeben ein Graph $G = (V, E)$ und ein Satz an Kantengewichtsmengen $W = W_1 \times \cdots \times W_N$, finde alle Kantengewichtskombinationen $\underline{w}_k$, die konsistent sind.*

Diese Synthese erfolgt zunächst über die Berechnung einer Basis des Maschenraumes, wie es die FM darstellen. Anschließend werden alle konsistenten Kantengewichtskombinationen für diese FM berechnet und auf zwei unterschiedliche Weisen zusammengefügt. Es werden dabei nur konsistente FM kombiniert, die gleiche Kantengewichte auf gemeinsamen Kanten haben. Die zwei Kombinationsverfahren haben folgende Zielsetzung:

1. Das Zusammenfügen führt zu einem gültigen Kantengewicht auf jeder Kante des Eingangsgraphen. Der resultierende Graph heißt *voll konsistent*.
2. Das Zusammenfügen findet alle konsistenten Teilgraphen, wobei ein konsistenter Teilgraph nur eine konsistente Masche oder sogar ein voll konsistenter Graph – sofern er existiert – sein kann. Dabei werden alle *partiell konsistenten* Lösungen gefunden.

Gegeben der zusammenhängende, gewichtete Eingangsgraph aus Abb. 4.1, so findet die Synthese voll konsistenter Graphen nur den einen konsistenten Graphen, der allen Kanten des Eingangsgraphen ein gültiges Kantengewicht zuordnet. Die Synthese partiell konsistenter Graphen findet hingegen alle möglichen, konsistenten Teilgraphen, die wie in diesem Beispiel aus nur einer Masche bestehen können, und den voll konsistenten Graphen.

Die Auswahl des geeigneten Verfahren ist von der jeweiligen Anwendung abhängig. Kann sichergestellt werden, dass die gesuchten Kantengewichte auf jeden Fall in der Kantengewichtsmenge enthalten sind und eine große Menge falscher Werte gemessen

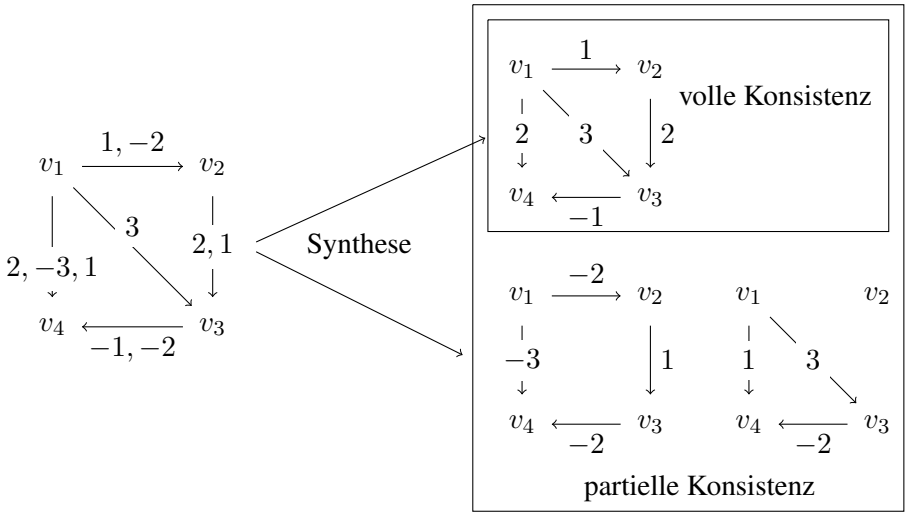


Abbildung 4.1: Ein gerichteter und gewichteter Eingangsgraph (links) und die Ergebnisse aus der Synthese voll konsistenter Graphen und der Synthese partiell konsistenter Graphen

werden, so ist die Synthese voll konsistenter Graphen sinnvoll, da sie effizient alle fehlerhaften Lösungen aussortiert. Ist es bei der Anwendung jedoch ungewiss, ob alle Kantengewichtsmengen zuverlässig die richtigen Kantengewichte enthalten, so sollte die Synthese partiell konsistenter Graphen gewählt werden. Dieses Verfahren sucht zuverlässig alle Teillösungen.

Da bei der TDOA-basierten Lokalisierung die Schätzung der TDOA stark von umgebenden Quellen und den relativen Position der Quellen zu den Mikrofonen abhängt, kann es vorkommen, dass die Direktpfad-TDOA zu einer Quelle an einem Mikrofonpaar nicht oder nur schlecht geschätzt werden. In diesem Fall ist es besser, mit den restlichen Mikrofonpaaren einen konsistenten Graphen zu synthetisieren und mit weniger TDOA-Werten eine Lokalisierung durchzuführen. Sonst wird die Quelle anstatt ungenau gar nicht lokalisiert, was meistens der schlechtere Fall ist.

4.1 Ansätze für die Synthese

4.1.1 Effizientes Syntheseverfahren für voll konsistente Graphen

Nachdem aus der Komponentensuche die zweifach-zusammenhängenden Teilgraphen bestimmt wurden, werden diese sequentiell abgearbeitet. Dabei werden für jede Komponente ein Satz von FM berechnet und anschließend die Synthese gestartet. Ausgehend von allen FM werden zunächst für jede FM $\underline{l}_{f,i}$ alle Kantengewichtskombinationen

mit

$$\underline{l}_{f,i}^{T}\underline{w} = 0 \tag{4.1}$$

auf Konsistenz überprüft. Dabei müssen abhängig von den beteiligten Kanten in der FM $\prod_n K_n$ Kombinationen mit $n \in \{1 \leq n \leq N : \underline{l}_{f,i}(n) \neq 0\}$ geprüft werden. Daraus entstehen $\tilde{K}_i$ gewichtete, konsistente FM $\underline{l}_{f,i}^{w_k}$ $(1 \leq i \leq N_{\mathrm{FM}}, 1 \leq k \leq \tilde{K}_i)$. Diese unterscheiden sich von den bisherigen topologischen FM $\underline{l}_{f,i} \in \{-1, 0, 1\}^N$, da sie als gewichtete, konsistente FM auch die k-te konsistente Kantengewichtskombination $\underline{w}_k$ beinhalten und somit $\underline{l}_{f,i}^{w_k} \in \mathbb{R}^N$ gilt. Die konsistenten FM werden anschließend durch das Rückverfolgungsverfahren (Backtracking, BT) rekursiv zu einem voll konsistenten Graphen zusammengefügt [Dechter, 2003; Tack, 2009; Kreißig u. Yang, 2012].

Der zu einer topologischen FM $\underline{l}_{f,i}$ gehörige Graph wird im Folgenden als $F_i = (\tilde{V}, \tilde{E})$ beschrieben, mit $\tilde{E} = \{e_n \in E : \underline{l}_{f,i}(n) \neq 0\}$ und den Endknoten $\tilde{V}$ aller Kanten in $\tilde{E}$. Das Zusammenfügen $G = F_i \cup F_j$ entspricht der Vereinigung der Knoten- und Kantenmengen, $V(G) = V(F_i) \cup V(F_j)$ bzw. $E(G) = E(F_i) \cup E(F_j)$.

Für eine gewichtete, konsistente FM $\underline{l}_{f,i}^{w_k}$ ist der zugehörige konsistente Graph $F_i^{w_k} = (F_i, \underline{w}_k)$ mit dem Graphen F_i der FM $\underline{l}_{f,i}$ und der konsistenten Kantengewichtskombination $\underline{w}_k$ gegeben. Werden zwei konsistente Maschen $F_i^{w_l} = (V_i, E_i, \underline{w}_l)$ und $F_j^{w_o} = (V_j, E_j, \underline{w}_o)$ zusammengefügt, so gilt $G^{w_k} = F_i^{w_l} \cup F_j^{w_o}$ mit $G(G^w) = G(F_i^{w_l}) \cup G(F_j^{w_o})$ genau dann, wenn für die gemeinsamen Kanten $S = E_i \cap E_j$ die Kantengewichte gleich sind. Ist $\underline{w}_l(n) \neq \underline{w}_o(n)$ für $e_n \in S$, können die gewichteten Teilgraphen nicht zusammengefügt werden.

Das BT-Verfahren nimmt zunächst eine topologische Masche F_1 und davon die erste konsistente Kantengewichtsbelegung $\underline{w}_1$. Dann selektiert BT für die nächste FM F_2 eine passende konsistente FM z.B. $F_2^{w_1}$ wie in Abb. 4.2 skizziert. Wird für die topologische FM F_3 keine passende konsistente Kantengewichtskombination $\underline{w}_k$ gefunden, erfolgt ein Rückschritt zur nächst niederigeren Stufe und es wird für F_2 die nächste passende konsistente FM $F_2^{w_2}$ gesucht. Dann versucht BT wieder für die topologische FM F_3 eine passende Kantengewichtsbelegung zu finden. Dies geschieht über alle FM hinweg, bis für die letzte N_{FM}-te FM eine passende konsistente Gewichtskombination, $\underline{w}_x$ oder $\underline{w}_y$ in Abb. 4.2, gefunden worden ist. Erst dann wird ein voll konsistenter Graph als Lösung gespeichert und die Suche in einer niedrigeren FM fortgesetzt.

Der Pseudocode zum BT-Verfahren ist in Alg. 5 gezeigt und wird durch $\mathrm{BT}(1, \emptyset, \emptyset, \underline{0}_N)$ initial aufgerufen. Solange die letzte FM noch nicht erreicht ist, wird für alle konsistenten Kantengewichtskombinationen nach gemeinsamen Kanten mit gleichen Kantengewichten gesucht. Sind auf gemeinsamen Kanten unterschiedliche Gewichte, so wechselt BT zur nächsten Kantengewichtskombination $\underline{w}_k$. Ansonsten werden der bisherige konsistente Teilgraph $\tilde{G}^w$ und die aktuelle, konsistente FM $F_i^{w_k}$ zusammengefügt und BT mit der nächsten FM aufgerufen. Ist die letzte FM erreicht, speichert Alg. 5 die Lösung.

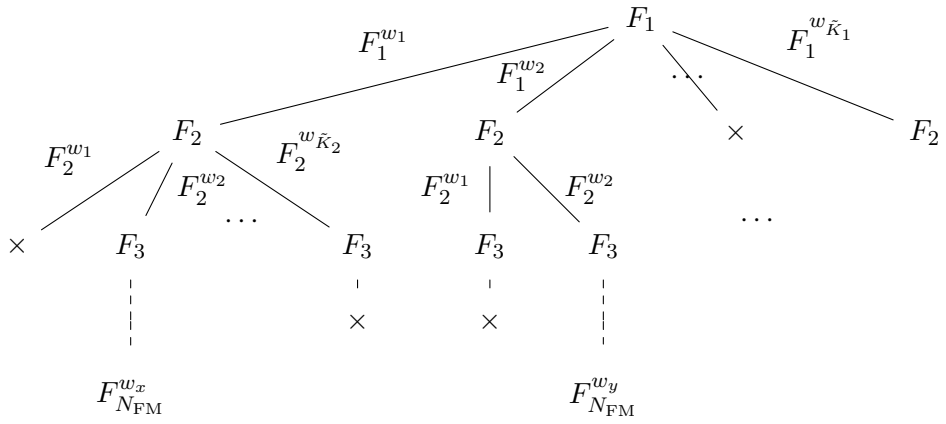


Abbildung 4.2: Das Zusammenführen der FM für die Synthese voll konsistenter Graphen. $\times$ zeigt einen Konflikt der bisherigen Lösung mit der aktuellen konsistenten FM auf. Gestrichelte Linien deuten auf eine vollständige Lösung hin

Nach Abschluss erhält man die $\hat{K}$ voll konsistenten Graphen $G^{w_k} = (G, \underline{w}_k)$ mit den konsistenten Kantengewichtskombinationen $\underline{w}_k,\ 1 \leq k \leq \hat{K}$.

Algorithmus 5: Pseudocode für die Synthese voll konsistenter Graphen

Input : BT($i, \tilde{V}, \tilde{E}, \underline{\tilde{w}}$), FM-Nummer i, konsistenter Teilgraph $\tilde{G}^w$
Output : Voll konsistente Graphen $G^{w_k}, 1 \leq k \leq \hat{K}$

```
if i ≤ N_FM then                                          // Für alle FM
    S = Ẽ ∩ E(F_i);                          // Finde gemeinsame Kanten zw. Teilgraph und i-ter FM
    for 1 ≤ k ≤ K̃_i do                  // Für alle konsistenten Kantengewichtskombinationen
        merge = 1;
        while s ∈ S do                                    // Für alle gemeinsamen Kanten
            if w̲̃(s) ≠ w̲_k(s) then                  // Suche unterschiedliche Kantengewichte
                merge = 0;
            end
        end
        if merge then                                     // Falls keine Widersprüche existieren
            G̃^w = (Ṽ, Ẽ, w̲̃) ∪ F_i^{w_k};              // Verbinde Teilgraph mit konsistenter FM
            BT(i + 1, V(G̃^w), E(G̃^w), w̲̃(G̃^w));          // Durchsuche nächste FM
        end
    end
else
    speichere G̃^w = (Ṽ, Ẽ, w̲̃);                        // G^{w_k} = G̃^w ist voll konsistent
end
```

Das BT-Verfahren eignet sich besonders für die Synthese voll konsistenter Graphen, da hierbei jeder Kante des Eingangsgraphen ein Kantengewicht zugeordnet werden muss. Da manche Kanten nur in einer FM enthalten sind, müssen alle FM eine passende Kantengewichtskombination haben. D.h. sobald keine passende Kantengewichtskombination für eine FM gefunden ist, kann die weitere Synthese abgebrochen werden und die vorherige FM muss eine andere Kantengewichtskombination wählen. Diese Effizienz ist auch in Abb. 4.2 erkennbar, da ganze Zweige von unnötigen Kombinationsmöglichkeiten abgeschnitten werden, weil sie zu keiner voll konsistenten Lösung führen.

In den meisten realen Anwendungen kann es aber durch Messfehler oder niedrige Signalleistungen dazu kommen, dass nicht alle Kantengewichtsmengen W_n die richtigen Direktpfad-TDOA enthalten. In diesen Fällen wird die Synthese voll konsistenter Graphen mittels des BT-Verfahrens die gewünschte Lösung verlieren, da alle Kantengewichte im Eingangsgraphen enthalten sein müssen. Deshalb ist es für die allgemeine Synthese wichtig, auch konsistente Teilgraphen zu finden, in denen ein oder mehrere Kantengewichte fehlen, aber die restlichen Kanten gültig belegt sind. Diese Synthese partiell konsistenter Graphen wird im folgenden Kapitel betrachtet.

4.1.2 Syntheseverfahren für partiell konsistente Graphen

Ausgehend von den konsistenten FM $\underline{l}_{f,i}^{w_k}$ aus (4.1) werden alle möglichen Kombinationen von konsistenten FM gesucht, wie es in Abb. 4.3 gezeigt ist. Dort sind die drei topologischen FM aus Abb. 3.6 b) mit jeweils zwei konsistenten Kantengewichtskombinationen dargestellt. Man kann direkt erkennen, dass sich $F_1^{w_1}$ mit $F_2^{w_2}$ und $F_3^{w_1}$ kombinieren lassen, weil an den gemeinsamen Kanten die gleichen Kantengewichte vorliegen. Daraus folgt der erste, (voll) konsistente Graph $G^{w_1} = F_1^{w_1} \cup F_2^{w_2} \cup F_3^{w_1}$. Die weiteren partiell konsistenten Lösungen zu dem Beispiel aus Abb. 4.3 ergeben sich zu $G^{w_2} = F_1^{w_2} \cup F_2^{w_1}$ und $G^{w_3} = F_3^{w_2}$.

Die Synthese partiell konsistenter Graphen ist nicht trivial, da ein direkter Ansatz, alle konsistenten FM zu vergleichen mit $\prod_{i=1}^{N_{\mathrm{FM}}} \tilde{K}_i$ Kombinationsmöglichkeiten ineffizient ist. Der Algorithmus DATEMM [Scheuing u. Yang, 2008] ist hierfür eine mögliche Lösung und wird nachfolgend genauer betrachtet. Weil DATEMM aus mathematischer Sicht ein paar Fehler aufweist, wird anschließend in Kap. 4.2 ein vollständiger Ansatz vorgestellt.

DATEMM

Der Algorithmus DATEMM, wie er für die Synthese konsistenter TDOA Graphen eingeführt wurde, kann wie folgt zusammengefasst werden. Gesucht wird der konsistente Graph aus Abb. 4.4(a). DATEMM bestimmt zu diesem Graphen alle Maschen, die aus

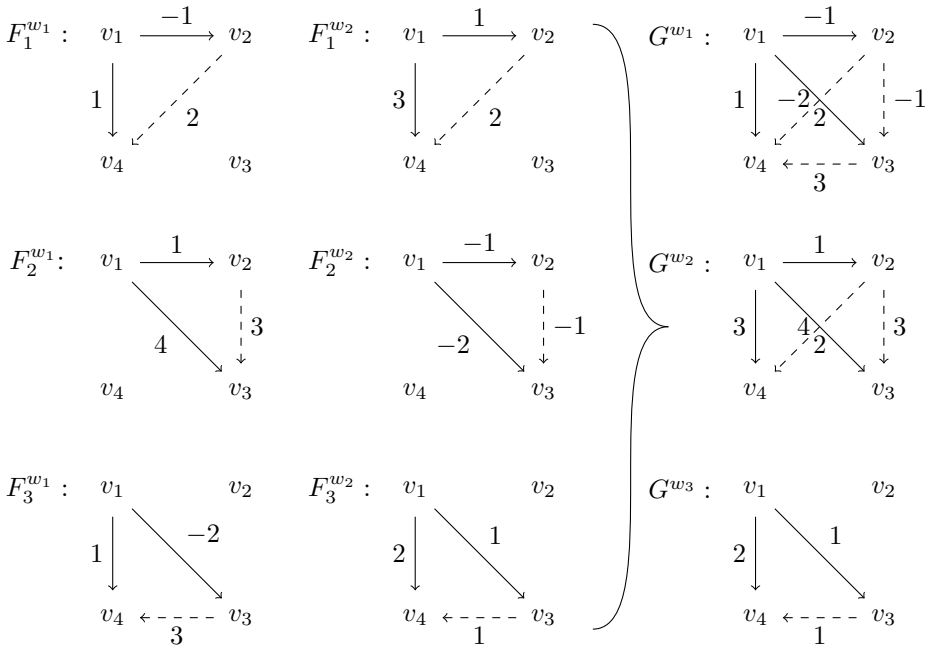
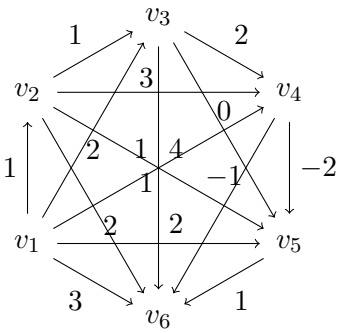


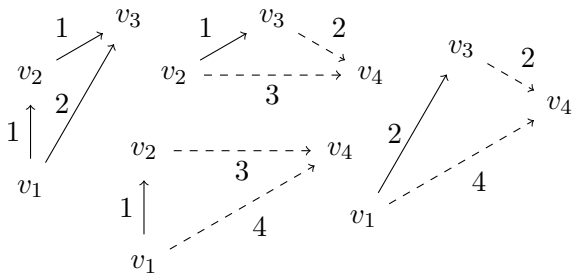
Abbildung 4.3: Konsistente Kantengewichtskombinationen für die FM aus Abb. 3.6

drei Kanten bestehen, und berechnet die konsistenten Kombinationen wie in Abb. 4.4(b) angedeutet. Aus den konsistenten Tripeln wird ein Starttripel gewählt, hier (v_1, v_2, v_3). Anschließend werden weitere konsistente Tripel angefügt und das konsistente Quadrupel aus Abb. 4.4(c) synthetisiert. Mehrere solcher konsistenter Quadrupel werden zu einem sogenannten Sterngraph zusammengefasst, wie er in Abb. 4.4(d) zu sehen ist. Ist das konsistente Tripel (v_4, v_5, v_6) vorhanden, wird dieses auch noch hinzugefügt. DATEMM gibt auch partiell konsistente Graphen zurück, wenn keine weiteren konsistenten Tripel zum Anfügen vorhanden sind. Wenn ein konsistenter Graph synthetisiert wurde, werden die benutzten konsistenten Tripel aus der Suchmenge entfernt und ein neues Starttripel gewählt.

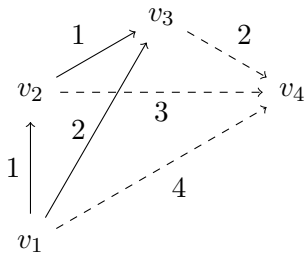
Dieser ad-hoc Ansatz weist ein paar Einschränkungen auf [Kreißig u. Yang, 2011]. Dazu zählt zum einen die feste Maschenlänge von drei Kanten. Diese Einschränkung verlangt, dass der zugrunde liegende Graph zusammenhängend und nahezu vollständig ist, damit möglichst alle Tripel enthalten sind. Das ist ein Nachteil, denn es kann nicht immer gewährleistet werden, dass alle Kanten gegeben sind. Die Wahl einer flexiblen Basis für die Maschen, wie es die FM darstellen, ist ein allgemeinerer und robusterer Ansatz. Zum anderen ist die Wahl des Starttripels in DATEMM kritisch. Hat dieses Tripel wenige passende Nachbarn, so wird der synthetisierte Graph klein. Das letzte und größte Problem ist aber das Entfernen aller an einer Lösung beteiligten konsistenten Tripel von der weiteren Synthese. Wie in Tab. 4.1 veranschaulicht, können die glei-



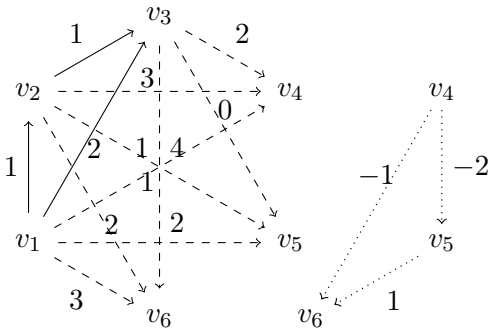
(a) Gesuchter Graph



(b) Die konsistenten Tripel zwischen den Knoten v_1, v_2, v_3 und v_4



(c) Quadrupel aus den Tripel von Abb. 4.4(b).



(d) Synthetisierter Sterngraph aus den konsistenten Quadrupeln und das fehlende Tripel

Abbildung 4.4: Syntheseschritte des DATEMM-Verfahrens

chen Tripel in unterschiedlichen Lösungen vorkommen. Sollte nun schlimmstenfalls $\underline{w}_3$ synthetisiert werden, so entfällt für $\underline{w}_1$ das Tripel (v_1, v_2, v_6) und für $\underline{w}_2$ das Tripel (v_3, v_4, v_5). Damit werden diese Lösungen nie ganz gefunden.

	w_{12}	w_{13}	w_{14}	w_{15}	w_{16}	w_{23}	w_{24}	w_{25}	w_{26}	w_{34}	w_{35}	w_{36}	w_{45}	w_{46}	w_{56}
$\underline{w}_1^T$	[1]	−2	1	2	[−2]	−3	0	1	[−3]	3	4	0	1	−3	−4
$\underline{w}_2^T$	2	3	4	5	3	1	2	3	1	(1)	(2)	0	(1)	−1	−2
$\underline{w}_3^T$	[1]	1	2	3	[−2]	0	1	2	[−3]	(1)	(2)	−3	(1)	−4	−5

Tabelle 4.1: Beispiel dreier Graphen, die von DATEMM nicht gleichzeitig gefunden werden

Da die Synthese nicht entscheiden kann, welche Lösungen richtig oder falsch sind, müssen alle Kombinationen betrachtet werden, was mit dem Ausschluss der bereits verwendeten Tripel nicht möglich ist. Deshalb wurde ein Suchverfahren entwickelt, dass

möglichst effizient die passenden Kombinationen sucht, ohne eine Teilkombination zu verlieren. Dieses Verfahren wird Kap. 4.2 eingeführt. Vorher wollen wir noch untersuchen, ob auch eine andere Basis als die FM für die Synthese möglich ist.

4.1.3 Fundamentale oder unabhängige Maschen

Die Frage, die uns an dieser Stelle beschäftigt, ist, ob auch andere unabhängige Maschenvektoren existieren, die anstelle der FM für die Konsistenzprüfung und somit für die Synthese konsistenter Graphen eingesetzt werden können. Wie in [Yang u. Kreißig, 2013] gezeigt, ist die Analyse der Konsistenz mittels einer beliebigen Menge unabhängiger Maschen nach (3.6) möglich, da ein konsistenter Graph immer eine Nullsumme entlang aller Maschen aufweist. Die Umkehrung, dass die Synthese durch Zusammenfügen von beliebigen, unabhängigen, konsistenten Maschen wiederum zu einem konsistenten Graphen führt, gilt allerdings nicht, wie Abb. 4.5 zeigt.

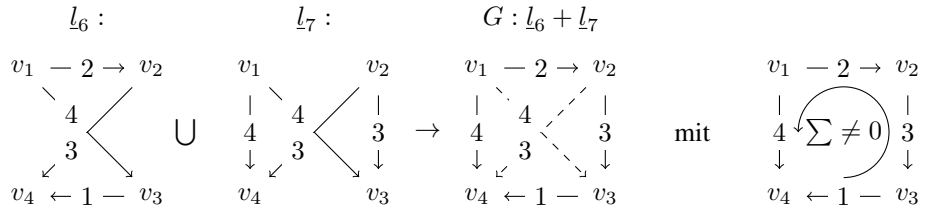


Abbildung 4.5: Eine schlechte Basis linear unabhängiger Maschen für die Synthese konsistenter Graphen

Die Maschen $\underline{l}_6$ und $\underline{l}_7$ entstammen der Maschenmatrix aus (3.2) und sind sogar orthogonal, wie $\underline{l}_6^T \underline{l}_7 = 0$ zeigt. Zusammen mit $\underline{l}_1$ ergeben sie die Maschenmatrix

$$\mathbf{L} = \begin{bmatrix} & \underline{l}_1 & \underline{l}_6 & \underline{l}_7 \\ e_1 & 1 & 1 & 0 \\ e_2 & -1 & -1 & 1 \\ e_3 & 0 & 0 & -1 \\ e_4 & 1 & 0 & -1 \\ e_5 & 0 & 1 & 1 \\ e_6 & 0 & -1 & 0 \end{bmatrix} \tag{4.2}$$

und sind eine linear unabhängige Basis des 3-dimensionalen Maschenraums des 6-dimensionalen Kantenraums, wie durch das Gaußsche Eliminationsverfahren gezeigt werden kann. Trotz der linearen Unabhängigkeit und der Konsistenz der Maschen $\underline{l}_6$ und $\underline{l}_7$ ist der zusammengefügte Graph G bzw. die sich aus der symmetrischen Differenz ergebende Masche nicht konsistent. Deshalb werden im Folgenden der Unterschied von FM und einer beliebigen Basis unabhängiger Maschen genauer betrachtet.

Lemma 1. *Gegeben zwei FM mit den Kantenmengen E_i und E_j vom gleichen aufspannenden Baum. Für die nicht-leere Menge $S = E_i \cap E_j$ gilt, dass alle Kanten in S zusammenhängend sind.*

Beweis. Da die FM über einen Pfad von den beiden Endknoten einer Kobaumkante entlang des aufspannenden Baumes berechnet werden, liegen alle möglichen gemeinsamen Kanten im aufspannenden Baum. Da ein aufspannender Baum azyklisch ist und immer nur ein Pfad zwischen zwei Knoten existiert, gilt für mindestens zwei gemeinsame Kanten, dass alle dazwischen liegenden Kanten eingeschlossen sind. Somit ist $E_i \cap E_j$ zusammenhängend. □

Die Maschen aus Abb. 4.5 sind wegen $\underline{l}_6^T \underline{l}_7 = 0$ unabhängig, erfüllen aber nicht die Eigenschaft, dass ihre gemeinsamen Kanten $\{v_1, v_3\}$ und $\{v_2, v_4\}$ zusammenhängend sind.

Lemma 2. *Zwei FM $\underline{l}_{f,i}$ und $\underline{l}_{f,j}$ stellen eine Basis des Maschenraumes des entstehenden Graphen $G = F_i \cup F_j$ dar.*

Beweis. Aus Lemma 1 folgt, dass $\underline{l}_{f,i}$ und $\underline{l}_{f,j}$ genau $s = |S|$ gemeinsame Kanten und $s+1$ gemeinsame Knoten haben. Hieraus ergibt sich für G, dass die Kantenanzahl $N = N_i + N_j - s$ und die Knotenanzahl $M = M_i + M_j - (s+1)$ ist, mit $N_i = |E(F_i)|$ und $M_i = |V(F_i)|$ und ebenso für F_j. Das bedeutet für den entstehenden Maschenraum von G, dass $N_{\mathrm{FM}} = (N_i+N_j-s)-(M_i+M_j-(s+1))+1 = (N_i-M_i+1)+(N_j-M_j+1)$ ist und genau der Anzahl der einzelnen Ränge entspricht. Zudem sind alle $\underline{l}_{f,i}$ und $\underline{l}_{f,j}$ unabhängig aufgrund ihrer eigenen Kobaumkante. Somit spannen $\underline{l}_{f,i}$ und $\underline{l}_{f,j}$ genau eine Basis für den Maschenraum von G auf. □

Für die unabhängigen Maschen $\underline{l}_6$ und $\underline{l}_7$ aus Abb. 4.5 ist $N_6 = 1$, $N_7 = 1$, aber $N_{\mathrm{FM}} = 3 \neq N_6 + N_7$. Damit spannen $\underline{l}_6$ und $\underline{l}_7$ keine Basis für G auf.

Theorem 1. *Nur die Kombination von konsistenten FM $F_i^{w_k}$ führt immer zu einem konsistenten Graphen.*

Beweis. Für einen beliebigen Graphen $G = (V, E)$ erfüllen alle FM Lemma 1 und Lemma 2. Damit ist jeder Zwischengraph $\tilde{G}^w = \cup_i F_i^{w_k}$, der aus dem Zusammenfügen einer beliebigen Anzahl von konsistenten FM $F_i^{w_k}$ besteht, ebenso konsistent, weil die Basismaschen des Graphen $\tilde{G}^w$ alle (3.6) erfüllen. □

Für allgemeine, unabhängige Maschen gilt Theorem 1 nicht, wie das Beispiel in Abb. 4.5 zeigt.

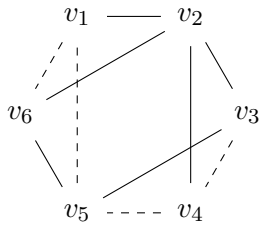
4.2 CC-Graph und CC-Graph-Suche

4.2.1 Einführung des CC-Graphs

Nachdem die konsistenten FM als Grundlage für die Synthese partiell konsistenter Graphen gewählt wurden, gilt es diese effizient zusammenzufügen. Das mögliche Zusammenführen von zwei konsistenten FM kann durch folgende drei Zustände charakterisiert werden:

Zustand	Zwei konsistente FM haben gemeinsame Kanten	Zwei konsistente FM haben an gemeinsamen Kanten gleiche Kantengewichte
kompatibel	ja	ja
in Konflikt	ja	nein
offen	nein	-

Daraus ergibt sich eine neue Darstellung in dem Kompatibilitäts-Konflikt-Graph (Compatibility-Conflict-Graph, CC-Graph). Der CC-Graph ist definiert als $G_{cc} = (V_{cc}, E, \bar{E})$ mit den kompatiblen Kanten E und den Konfliktkanten $\bar{E}$. Die Knotenmenge V_{cc} repräsentiert alle konsistenten FM. Jede Kante zwischen zwei Knoten (FM) repräsentiert einen der drei möglichen Zustände. Die kompatiblen Kanten (durchgezogene Linien in Abb. 4.6) stehen dafür, dass ihre Endknoten keinen Widerspruch aufweisen. Das heißt für die konsistenten FM, dass sie zusammengeführt werden können. Die Konfliktkanten (gestrichelte Linien in Abb. 4.6) deuten auf einen Widerspruch der Endknoten hin, was für die konsistenten FM unterschiedliche Kantengewichte auf gemeinsamen Kanten bedeutet. Schlussendlich ist der offene Zustand durch eine fehlende Kante dargestellt, da die möglichen Endknoten keine gemeinsamen Kanten haben.



$$V_{cc} = \{v_1, \ldots, v_M\}$$
$$E = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_2, v_4\}, \{v_2, v_6\}, \{v_3, v_5\}, \{v_5, v_6\}\}$$
$$\bar{E} = \{\{v_1, v_5\}, \{v_1, v_6\}, \{v_3, v_4\}, \{v_4, v_5\}\}$$

Abbildung 4.6: Beispielhafter CC-Graph mit den kompatiblen Kanten E als durchgezogene Linien und den Konfliktkanten $\bar{E}$ als gestrichelte Linien. Der offene Zustand zwischen zwei Knoten wird durch die fehlende Kante repräsentiert.

4.2.2 Die CC-Graph-Suche

Ziel für die Synthese partiell konsistenter Graphen ist es, alle möglichen kompatiblen Kombinationen von CC-Graph-Knoten zu finden, ohne eine Lösung zu verlieren. Für das Beispiel aus Abb. 4.6 sind alle maximal großen Lösungen ohne jegliche Konflikte $\{v_1, v_2, v_3\}, \{v_1, v_2, v_4\}, \{v_2, v_3, v_5, v_6\}$ und $\{v_2, v_4, v_6\}$. In Alg. 6 stellen wir den CCGsearch-Algorithmus vor, der das Problem, alle kompatiblen Knotenmengen im CC-Graph zu finden, löst [Kreißig u. Yang, 2013]. Es werden dabei systematisch alle Knoten $V = V_{cc}$ besucht und von jedem Knoten ausgehend alle kompatiblen, adjazenten Knoten zu einer möglichen Lösung L erweitert. Stellt CCGsearch im Laufe einer Suche fest, dass die Lösung bereits von einem anderen Knoten aus gefunden wurde, so verwirft es diese, um keine doppelten Lösungen zu generieren, und fährt fort.

Algorithmus 6: Pseudocode der CC-Graph-Suche

Input : CCGsearch($V, E, \bar{E}, L, X$), Knotenmenge V, kompatible Kanten E, Konfliktkanten $\bar{E}$, aktuelle Lösungsmenge L, besuchte Knotenmenge X
Output : Menge der Lösungsknotenmengen $\{L\}$

```
1  if L = ∅ then                                   // erste Rekursionstiefe wird geprüft
2      N_V(L) = V
3  else
4      for v ∈ V do                                // überprüfe alle Knoten
5          if ∃l ∈ L : {l, v} ∈ Ē then             // ob Konflikt vorliegt
6              N̄_V(L) = N̄_V(L) ∪ v;
7          else if ∃l ∈ L : {l, v} ∈ E then        // oder nach Kompatibilität
8              N_V(L) = N_V(L) ∪ v;
9      end
10 end
11 if N_V(L) = ∅ then   // falls kein kompatibler Nachbar existiert, ist die Lösung maximal
12     save L;                                     // → speichern
13 else
14     V = V \ N̄_V(L) ;          // Konfliktnachbarn von der weiteren Suche ausschließen
15     for n ∈ N_V(L) \ X do     // alle Nachbarn, die nicht bereits besucht wurden
16         CCGsearch(V \ n, E, Ē, L ∪ n, X);
17         X = X ∪ n;
18     end
19 end
```

Der Algorithmus wird mit CCGsearch($V_{cc}, E, \bar{E}, \emptyset, \emptyset$) aufgerufen und setzt im ersten Durchlauf die Menge der kompatiblen Nachbarknoten $N_V(L)$ auf die gesamte Knotenmenge des CC-Graph (Zeile 2). Anschließend werden die adjazenten Konfliktknoten $\bar{N}_V(L)$ von der Knotenmenge V entfernt (Zeile 14), wobei diese im ersten Durchlauf $\bar{N}_V(L) = \emptyset$ ist. Dann wird die erste Rekursion aufgerufen, wobei die Lösungsmenge L um den Knoten n der kompatiblen Nachbarn erweitert wird und dieser Knoten von

der Knotenmenge V entfernt wird (Zeile 16). Nachfolgend wird die Menge der kompatiblen, adjazenten Knoten nach folgender Regel bestimmt

$$N_V(L) = \{v \in V | \exists l \in L : \{l, v\} \in E \wedge \nexists \iota \in L : \{v, \iota\} \in \bar{E}\}, \tag{4.3}$$

wonach nur solche Knoten aus V hinzugefügt werden, die mindestens mit einem Knoten aus der Lösungsmenge L kompatibel sind und mit keinem Knoten aus L eine Konfliktkante haben. Diese Menge ist abhängig von der aktuellen Knotenmenge V, die in jedem Durchgang reduziert wird. Gleichzeitig wird in jeder Rekursion die Menge der in Konflikt stehenden, adjazenten Nachbarn nach

$$\bar{N}_V(L) = \{v \in V | \exists l \in L \wedge \{l, v\} \in \bar{E}\} \tag{4.4}$$

berechnet. Demnach ist ein Knoten in $\bar{N}_V(L)$, wenn mindestens eine inzidente Konfliktkante zu einem Knoten in L existiert. Somit kann ein Knoten aus V nicht in beiden Menge $N_V(L)$ und $\bar{N}_V(L)$ gleichzeitig vorkommen.

Es werden nun – wie vorher – die Konfliktnachbarn von der Knotenmenge entfernt und für jeden kompatiblen Nachbarknoten CCGsearch erneut aufgerufen, wobei der aktuelle Nachbarknoten zur Lösungsmenge hinzugefügt und aus der Knotenmenge V entfernt wird. Sind in einer Rekursion einmal keine weiteren kompatiblen Nachbarn vorhanden (Zeile 11), so wird die aktuelle Lösungsmenge L als maximal abgespeichert.

Da CCGsearch von jedem Knoten aus eine Suche startet, würden ohne eine Erweiterung des Algorithmus die einzelnen Lösungen mehrfach gefunden werden. Deshalb fügt CCGsearch bei der Rückkehr aus einer Rekursion den aktuellen Knoten n zu der Menge bereits besuchter Knoten X hinzu (Zeile 17). Dieser Schritt ist notwendig, um bereits gefundene Lösungen zu detektieren. So wird bei der Schleife über alle kompatiblen Nachbarn (Zeile 15) anfangs noch die bereits besuchten Knoten entfernt um keine Kombination mehrmals zu untersuchen.

Für das Beispiel aus Abb. 4.6 ist der Ablauf von Alg. 6 in Tab. 4.2 gezeigt. Es sind die einzelnen Knotenmengen je Rekursion in jeweils einer Zeile zu sehen. Die Rekursionstiefe ist in römischen Zahlen gegeben und die Verzweigungen innerhalb einer Rekursion mit Buchstaben. Die gespeicherten Lösungen sind mit $\boxed{}$ gekennzeichnet. Man erkennt, dass genau die gewünschten Lösungen gefunden werden.

4.2.3 Vollständigkeit der CC-Graph-Suche

Nach der Einführung der CC-Graph-Suche im vorherigen Kapitel, werden wir nun zeigen, dass CCGsearch

- alle Lösungen findet,
- jede Lösung nur einmal findet und

Tiefe	n	L	X	V	$N_V(L)$	$\bar{N}_V(L)$	$V \setminus \bar{N}_V(L)$	$N_V(L) \setminus X$
0		$\emptyset$	$\emptyset$	1,2,3,4,5,6	1,2,3,4,5,6	$\emptyset$	1,2,3,4,5,6	1,2,3,4,5,6
i_a	1	1	$\emptyset$	2,3,4,5,6	2	5,6	2,3,4	2
ii_a	2	1,2	$\emptyset$	3,4	3,4	$\emptyset$	3,4	3,4
iii_a	3	1,2,3	$\emptyset$	4	$\emptyset$	-	-	-
iii_b	4	1,2,4	3	3	$\emptyset$	-	-	-
i_b	2	2	1	1,3,4,5,6	1,3,4,6	$\emptyset$	1,3,4,5,6	3,4,6
ii_a	3	2,3	1	1,4,5,6	1,5,6	4	1,5,6	5,6
iii_a	5	2,3,5	1	1,6	6	1	6	6
iv_a	6	2,3,5,6	1	$\emptyset$	$\emptyset$	-	-	-
iii_b	6	2,3,6	1,5	1,5	5	1	5	$\emptyset$
ii_b	4	2,4	1,3	1,3,5,6	1,6	3,5	1,6	6
iii_a	6	2,4,6	1,3	1	$\emptyset$	-	-	-
ii_c	6	2,6	1,3,4	1,3,4,5	3,4,5	1	3,4,5	5
iii_a	5	2,5,6	1,3,4	3,4	3	4	3	$\emptyset$
i_c	3	3	1,2	1,2,4,5,6	2,5	4	1,2,5,6	5
ii_a	5	3,5	1,2	1,2,6	2,6	1	2,6	6
iii_a	6	3,5,6	1,2	2	2	$\emptyset$	2	$\emptyset$
i_d	4	4	1,2,3	1,2,3,5,6	2	3,5	1,2,6	$\emptyset$
i_e	5	5	1,2,3,4	1,2,3,4,6	3,6	1,4	2,3,6	6
ii_a	6	5,6	1,2,3,4	2,3	2,3	$\emptyset$	2,3	$\emptyset$
i_f	6	6	1,2,3,4,5	1,2,3,4,5	2,5	1	2,3,4,5	$\emptyset$

Tabelle 4.2: Detailierter Ablauf der CC-Graph Suche aus Alg. 6 zum CC-Graphen aus Abb. 4.6

- jede Lösung maximal ist.

Um dem Problem von doppelten Lösungen entgegen zu gehen, die sich aus der obigen Beschreibung ergeben würden, speichert der Algorithmus die bereits abgearbeiteten Knoten in X. Zu einem späteren Zeitpunkt erfolgt eine Abfrage, die aus $N_V(L)$ die bereits besuchten Knoten löscht.

Lemma 3. *CCGsearch findet zu einer gegebenen Lösungsmenge L alle kompatiblen Knotenmengen.*

Beweis. Es existieren zwei Wege, wie ein Knoten zu L kompatibel sein kann: direkt, wenn er ein adjazenter Knoten bezüglich E ist, oder indirekt, wenn er über einen Pfad von kompatiblen Knoten hinzugefügt wird. Im ersten Fall wird der Knoten gefunden und angefügt, wenn alle kompatiblen Nachbarn $N_V(L)$ durchsucht werden. Im zweiten Fall betrachtet man den rekursiven Aufruf von CCGsearch. Dabei wird L je Aufruf um einen Knoten erweitert. Dies schließt die Knoten aus dem verbindenden Pfad mit ein. Somit findet CCGsearch alle maximal und kompatibel zusammenhängende Knotenmengen zu L. □

Aus Lemma 3 ist bekannt, dass keine Lösung verloren geht und dass die Lösungen maximal sind. Lemma 4 zeigt, dass jede Lösung auch einmal gefunden wird.

Lemma 4. *CCGsearch findet jede Lösungsmenge L einmal, d. h. keine Lösung ist eine Teilmenge einer anderen Lösung.*

Beweis. Gegeben eine Lösungsmenge L, die mit den zwei gegenseitig offenen Knoten m und n kompatibel ist, so werden nach Lemma 3 alle Lösungen mit m und n gefunden. Existiert ein Knoten v, der sowohl mit m als auch mit n kompatibel ist und angenommen, dass m zuerst abgearbeitet wird, dann findet CCGsearch zuerst die Lösung $L \cup m \cup v \cup n$. Wenn diese Lösung maximal ist, wird sie gespeichert und CCGsearch geht zurück und m wird an X angefügt. Als nächstes findet CCGsearch die Lösung $L \cup n \cup v$. Weil im nächsten Aufruf $N_V(L)$ mindestens m enthält, wird diese Lösung nicht gespeichert. Bevor m aber an die Lösung gefügt wird, werden alle Knoten aus X von $N_V(L)$ entfernt. Dies beinhaltet auch m, wodurch die Lösung $L \cup n \cup v \cup m$ nicht nochmals gespeichert wird. □

4.2.4 Vergleich mit dem Konfliktgraph

Die CC-Graph-Suche CCGsearch ähnelt der Cliquensuche von [Bron u. Kerbosch, 1973], die sich mit der Suche nach vollständig zusammenhängenden Teilgraphen beschäftigt. Die Cliquensuche ist notwendig für Anwendungen wie z.B. das Rucksackproblem [Pferschy u. Schauer, 2009; Eisenbrand u. a., 2005] bei dem die Knoten des Graphen kleine Pakete darstellen, mit denen ein Raum (Rucksack) bestmöglich gefüllt werden soll. Dabei besteht die Einschränkung, dass zwei Pakete sich nicht überschneiden dürfen. Falls sie sich überschneiden, werden ihre entsprechenden Knoten im *Konfliktgraph* verbunden. Somit stellt der Konfliktgraph alle Widersprüche zwischen den einzelnen Knoten (Pakete mit der jeweiligen Position und Ausrichtung) dar. Um nun eine Lösung für das bestmögliche Füllen des Raumes mit möglichst vielen Paketen zu erhalten, sucht man im Konfliktgraph nach Knotenmengen, die vollständig konfliktfrei sind, d.h. keine Kanten enthalten. Diese Knotenmengen werden auch *unabhängige* oder *stabile Mengen* genannt. Algorithmen hierfür sind u.a. von [Tarjan u. Trojanowski, 1976] und [Rebennack, 2009] gegeben.

Die unabhängigen Mengen entsprechen genau den Cliquen im Komplementgraph des Konfliktgraphs. Der Komplementgraph enthält dabei nur die Kanten, die im Konfliktgraph nicht enthalten sind. Somit ist die Suche nach unabhängigen Knotenmengen zu der Suche nach Cliquen im Komplementgraph identisch. Neben der Cliquensuche nach [Bron u. Kerbosch, 1973] ist noch [Akkoyunlu, 1973] zu nennen.

Die Lösung des CC-Graph-Problems über den Konfliktgraph teilt sich in drei Phasen:

1. Finde alle konfliktfreien Knotenmengen mit der Cliquensuche im Komplement dieses Konfliktgraphs $G_c = (V_{cc}, \bar{E})$
2. Suche innerhalb der Lösungen nach kompatibel zusammenhängenden Knotenmengen
3. Entferne doppelte Lösungen oder Teillösungen

Teil 1 Gegeben den Konfliktgraph $G_c = (V_{cc}, \bar{E})$, so wird zuerst der Komplementgraph berechnet. Dabei wird ein vollständig zusammenhängender Graph $\bar{G}_c$ mit den V_{cc} Knoten erstellt und anschließend alle Kanten $\bar{E}$ davon entfernt. Die sich ergebenden Kantenmenge $E(\bar{G}_c)$ ist dabei unterschiedlich zu der Kantenmenge der kompatiblen Kanten $E(G_{cc})$ des CC-Graph. Anschließend müssen innerhalb des Graphen $\bar{G}_c$ alle Cliquen gefunden werden. Der Algorithmus zur Cliquensuche nach [Bron u. Kerbosch, 1973] ist in Alg. 7 dargestellt und der Aufruf erfolgt mit CliqueSearch($V(\bar{G}_c), \emptyset, \emptyset$). In der Menge $N(v)$ sind die zu v adjazenten Knoten enthalten, die direkt aus $E(\bar{G}_c)$ bzw. der Adjazenzliste abgelesen werden können und deren Berechnung deshalb nicht explizit genannt wird.

Algorithmus 7: Pseudocode der Cliquensuche nach Bron und Kerbosch

Input : CliqueSearch(V, L, X), Knotenmenge V, Lösungsmenge L, besuchte Knotenmenge X
Output : Menge der maximalen Cliquen $\{L\}$
if $V = \emptyset \wedge X = \emptyset$ **then** *// keine Knoten mehr, die noch nicht besucht wurden*
    save L; *// Clique ist maximal*
else
    for $v \in V$ **do** *// nur die adjazenten Knoten von v*
        CliqueSearch($V \cap N(v), L \cup v, X \cap N(v)$); *// in V und X werden weitergegeben*
        $V = V \setminus v$;
        $X = X \cup v$;
    end
end

Solange noch Knoten in V vorhanden sind, wird CliqueSearch seine Lösungsmenge sukzessiv erweitern, wobei nur solche Knoten in der nachfolgenden Rekursion berücksichtigt werden, die mit dem aktuellen Knoten v verbunden sind, $V \cap N(v)$. Dadurch wird der vollständige Zusammenhang gewährleistet, da alle Knoten in V mit allen Knoten in L verbunden sind. Hat CliqueSearch alle Knoten besucht, wird die Lösung als maximale Clique gespeichert. Anschließend werden die bereits besuchten Knoten v von der Knotenmenge entfernt und der Menge der bereits besuchten Knoten X hinzugefügt. Dieser Mechanismus gewährleistet, wie bei CCGsearch, dass keine Cliquen doppelt gefunden werden, da am Ende einer Cliquensuche immer die Knotenmenge V und die

Menge der besuchten Knoten X betrachtet werden und nur wenn beide Mengen leer sind, wird eine Lösung als neue Clique abgespeichert.

Teil 2 Sind alle Cliquen gefunden, müssen diese im zweiten Schritt nachbearbeitet werden, da bisher nur die Konfliktfreiheit sichergestellt wurde. Es kann sein, dass die Knoten aus einer Clique keine kompatible Verbindung in $E(G_{cc})$ aufweisen. Für die Synthese konsistenter Graphen bedeutet das, dass zwei konsistente FM keine gemeinsamen Kanten im Eingangsgraphen $G = (V, E)$ aufweisen und somit auch kein Konflikt besteht. Diese offenen Cliquenknoten-Paare im CC-Graph sind in der Synthese als separate Lösungen zu betrachten und müssen somit noch gefunden und getrennt werden.

Welche Knoten in einer Clique $\mathcal{V}$ kompatibel verbunden sind, kann über eine Baumsuche in $(\mathcal{V}, E(G_{cc}))$ mit den kompatiblen Kanten des CC-Graph erfolgen.

Teil 3 Sind alle kompatiblen, konfliktfreien Knotenmengen des CC-Graphs gefunden, folgt ein finaler Schritt, um doppelte Lösungen zu detektieren. Durch das Teilen der Cliquen aus Alg. 7 kann es zu Wiederholungen kommen, wie das folgende Beispiel zeigt: Gegeben den Fall, dass die konsistente FM $F_1^{w_k}$ sowohl in der Lösung $L^{w_1} = F_1^{w_k} \cup F_2^{w_l}$ als auch in $L^{w_2} = F_1^{w_k} \cup F_3^{w_m}$ enthalten ist. Ist weiterhin mindestens eine dieser Lösungen nicht kompatibel zusammenhängend (o.B.d.A L^{w_1}), so wird die Nachverarbeitung $F_1^{w_k}$ als Einzellösung zurückgegeben. Damit ist $F_1^{w_k} \subset L^{w_2}$ keine maximale Lösung oder falls L^{w_2} auch nicht kompatibel zusammenhängend ist, ist $F_1^{w_k}$ doppelt und keine einmalige Lösung mehr.

Durch die genannten drei Arbeitsschritte ist das Lösen der CC-Graph-Suche mit dem Konfliktgraph sehr ineffizient, da viele redundante Lösungen generiert werden. Stattdessen bietet CCGsearch einen direkten und geschlossen Ansatz und ist deshalb zu bevorzugen.

4.3 Überblick des SONG-Algorithmus

Die in den vorangegangenen Kapiteln im Detail diskutierten Schritte sind in Abb. 4.7 zusammengefasst und stellen zusammen die Synthese konsistenter Graphen (synthesis of consistent graphs, SONG) dar. Zuerst erfolgt die Komponentensuche nach dem Verfahren von [Hopcroft u. Tarjan, 1973] und findet die zweifach-zusammenhängenden Teilgraphen. Da die nachfolgenden Schritte auf die Komponenten angewendet werden, wählen wir für den Eingangsgraphen die Notation $G_{\text{tot}} = (V_{\text{tot}}, E_{\text{tot}})$ mit $M_{\text{tot}} = |V_{\text{tot}}|$ und $N_{\text{tot}} = |E_{\text{tot}}|$ und für die Komponenten die bisher bekannte Notation $G = (V, E)$. Die anschließende Berechnung der FM wird nach dem in [Balabanian u. Bickart, 1969] beschriebenen Verfahren durchgeführt. Dabei wird zunächst ein aufspannender

Baum mittels der BFS-Suche berechnet, der, wie später ersichtlich wird und bereits in [Kreißig u. Yang, 2012] erläutert, zu einem effizienteren Verfahren führt. Außerdem verwendet der SONG-Algorithmus den Knoten mit der höchsten Gradzahl als den Wurzelknoten für die Suche nach dem aufspannenden Baum. Diese Einstellungen führen zu kurzen FM, die aufgrund der geringeren Anzahl von Kantengewichten in der zyklischen Summenberechnung weniger fehleranfällig sind und schneller verarbeitet werden können.

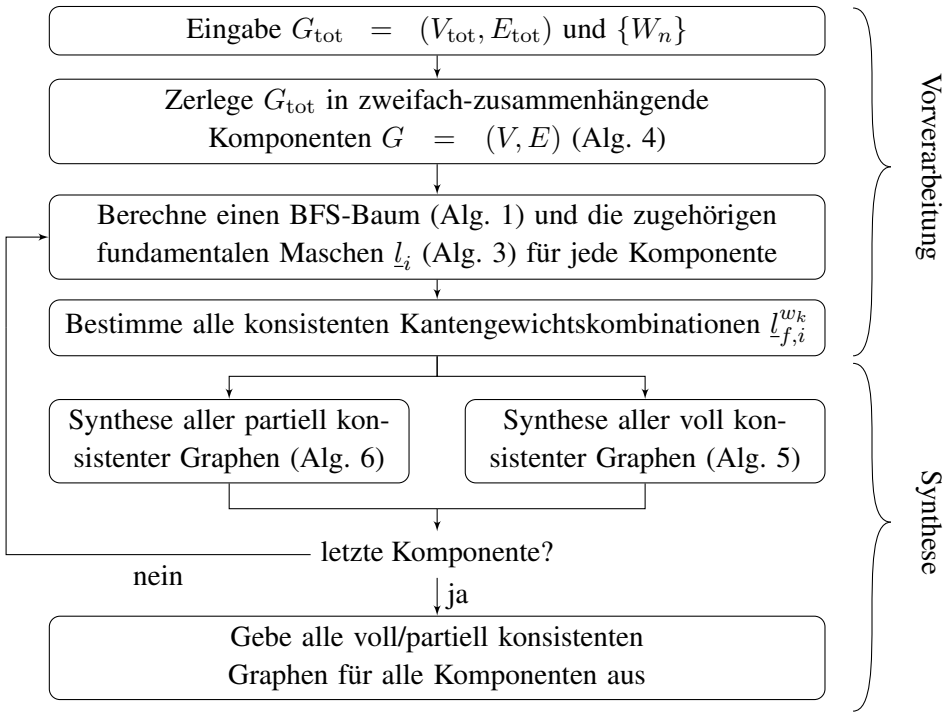


Abbildung 4.7: Blockdiagramm der Synthese konsistenter Graphen (SONG)

Die FM werden einzeln auf Konsistenz überprüft und als konsistente FM $\underline{l}_i^{w_k}$ gespeichert ($1 \leq i \leq N_{\mathrm{FM}}$ und $1 \leq k \leq \tilde{K}_i$). Anschließend folgt die Kombination aller konsistenten FMs zu voll oder partiell konsistenten Graphen. Da SONG von den kleinsten konsistenten Teilgraphen – der konsistenten FM – zum größtmöglichen konsistenten Graphen synthetisiert, wird er auch als Bottom-up-Synthese bezeichnet. Weil sowohl die Synthese voll konsistenter Graphen als auch die Synthese partiell konsistenter Graphen je nach Anwendungen von Vorteil sind, werden beide Verfahren in SONG implementiert.

4.4 Effizienzanalyse

4.4.1 Definition der Komplexität

Um die Komplexität eines Verfahrens genauer zu spezifizieren, bricht man es auf seine Grundrechenoperationen herunter, die von einem Prozessor in einem Arbeitsschritt abgearbeitet werden können. Die betrachteten Operationen in dieser Arbeit sind Vergleiche, Vereinigungen und Entnahme von Elementen aus einer Menge. Als Eingabeparameter für die Komplexität dient immer die Länge der Eingangsgröße wie z. B. die Anzahl der Abtastwerte für die schnelle Fourier-Transformation oder die Anzahl von Stationen (Knoten) bei Problem auf Graphen.

Neben der Komplexität der zeitlichen Dauer, allgemein als *TIME* benannt, ist auch die Komplexität der Speichernutzung *SPACE* ein wichtiger Indikator für die Effizienz eines Verfahrens. In der Informatik werden die Komplexitäten in Klassen eingeteilt, die durch einen maximalen Funktionstyp beschrieben werden.

Definition 2. *Ein Algorithmus der Komplexität $c(x)$ ist in der Komplexitätsklasse von $f(x)$, geschrieben $c(x) \in \mathcal{O}(f(x))$, wenn ein x_0 existiert, so dass für alle $x > x_0$: $c(x) < \alpha f(x)$ mit einem beliebigen Skalar $\alpha \in \mathbb{R} \setminus \{0, \infty\}$.*

Des weiteren ist nur noch ausschlaggebend, welche Funktion $f(x)$ beschreibt. Hierfür sind die bekanntesten Funktionen in Tab. 4.3 gegeben.

Funktion			Klassenname
polynomiell:	$f(x) = x^a$	$a \in \mathbb{R}$	*PTIME*
logarithmisch:	$f(x) = \log x$		*LOGTIME*
exponentiell:	$f(x) = a^x$	$a \in \mathbb{R}$	*EXPTIME*

Tabelle 4.3: Die bekanntesten Komplexitätsklassen. Mehr sind unter [Komplexitätszoo, 2014] zu finden

In der Informatik wird noch zwischen *deterministischen* und *nicht-deterministischen* Komplexitätsklassen unterschieden [Cormen u. a., 2009]. Gemeint sind damit die Klassen, die auf einer deterministischen Maschine (*PTIME* bzw. *P*) oder auf einer nicht-deterministische Maschine (*NTIME* bzw. *NP*) in polynomieller Zeit berechnet werden. Der Determinismus bezieht sich dabei auf die Vorschrift bei einem gegebenen Zustand der Maschine zum Nächsten zu wechseln. Bei den deterministischen Maschinen sind alle Übergänge fest vorgeschrieben. Bei den nicht-deterministischen Maschinen sind diese Übergänge nicht mehr eindeutig festgelegt. Es existieren mehrere Möglichkeiten und die nicht-deterministische Maschine wählt aus den parallelen Übergängen per Zufall den Richtigen aus. Bei der nicht-deterministischen Maschine handelt es sich um ein theoretisches Modell, welches nicht realisierbar ist. Deshalb gilt für Algorithmen

in *NP* auch weithin die schwächere Definition, dass ihre Lösung in polynomieller Zeit überprüft werden kann, was von der Rechenzeit für die Lösung unabhängig ist.

Ist es möglich zu zeigen, dass kein Algorithmus in polynomieller Zeit zu einem Problem existiert, so ist dieses Problem in der Klasse *NP-schwer*. Offensichtlich ist jedes Problem in *P* auch in *NP* ($P \subseteq NP$). Die Frage, ob *P* eine echte Teilmenge ist oder $P = NP$ gilt, ist noch nicht abschließend geklärt [Cormen u. a., 2009].

Ein Algorithmus, der in polynomieller Zeit auf einer nicht-deterministischen Maschine gelöst wird, kann ebenso auf einer deterministischen Maschine in exponentieller Zeit gelöst werden. Folgender Zusammenhang der Klassen ist schnell ersichtlich:

$$LOGTIME \subseteq PTIME \subseteq EXPTIME$$

4.4.2 Komplexität der Vorverarbeitung

Komplexität der Komponentensuche Die Suche nach zweifach-zusammenhängenden Komponenten ist ähnlich wie die Suche nach einem DFS aufspannenden Baum, weil alle Knoten und Kanten maximal einmal besucht werden. Diese Komplexität ist nach [Tarjan, 1972] gegeben als

$$C_{\mathrm{bi}} \in \mathcal{O}\left(M_{\mathrm{tot}} + N_{\mathrm{tot}}\right), \tag{4.5}$$

wobei M_{tot} die Knotenanzahl und N_{tot} die Kantenanzahl des Eingangsgraphen sind.

Komplexität des aufspanneden Baumes Die Komplexität für die Berechnung eines aufspannenden BFS- oder DFS-Baumes ist gegeben in [Cormen u. a., 2009] als

$$C_{\mathrm{st}} \in \mathcal{O}\left(M + N\right), \tag{4.6}$$

wobei M die Knotenanzahl und N die Kantenanzahl des zweifach-zusammenhängenden Teilgraphen sind.

Komplexität zum Schließen der FM Anschließend muss im aufspannenden Baum für jede Kante aus dem Kobaum ein Pfad gefunden werden, um je eine FM zu schließen. Dafür werden maximal

$$C_{\mathrm{fm}} = \mathcal{O}\left((N - M + 1)(2M - 1)\right) \tag{4.7}$$

Operationen benötigt, wenn für jede FM maximal $M - 1$ Kanten und M Knoten des aufspanneden Baumes durchsucht werden. Die Auswirkung des aufspannenden Baumes wird erst bemerkbar, wenn die Länge der resultierenden FM betrachtet wird.

Komplexität der Suche nach konsistenten FM Es werden genau $\prod_{n \in D_i} K_n \leq K^{N_i}$ Kantengewichtskombinationen je FM i auf die Nullsummenbedingung hin überprüft mit $D_i = \{1 \leq n \leq N : \underline{l}_{f,i}(n) \neq 0\}$, $K = \max_n K_n$ und $N_i = |D_i|$. Für einen vollständig zusammenhängenden Graphen gilt, dass ein BFS-Baum von seinem Startknoten eine Kante zu jedem weiteren Knoten aufweist. Fügt man nun eine Kobaumkante hinzu, so entsteht immer eine FM der Länge drei. Somit ist $N_i = 3$ und die Komplexität für die Suche nach den konsistenten FM ist

$$C_{\text{cfm}} \in \mathcal{O}\left((N - M + 1)K^3\right). \tag{4.8}$$

Im Vergleich dazu generiert der aufspannende DFS-Baum in einem vollständig zusammenhängenden Graphen genau eine FM der Länge M. Um alle FM auf Konsistenz zu überprüfen, muss demnach eine Kantengewichtskombination der Mächtigkeit $\prod_{n=1}^{M} K_n$ überprüft werden. Für nicht vollständig zusammenhängende Graphen heben sich die Unterschiede auf, weil dann die FM des BFS-Baumes länger und die des DFS-Baumes kürzer werden. Generell kann aber festgehalten werden, dass BFS zu weniger Kantengewichtskombinationen führt und somit effizienter ist.

Für die Länge der FM ist nicht nur die Suchstrategie des aufspannenden Baumes entscheidend, sondern auch die Wahl des Startknotens. Um eine geringe Länge zu gewährleisten, muss der Startknoten viele Nachbarn besitzen. Dieser Parameter lässt sich über den Knotengrad ausdrücken. Für die Synthese wird deshalb ein Knoten mit dem höchsten Knotengrad als Wurzelknoten für die Berechnung des aufspannenden Baumes gewählt.

4.4.3 Komplexität der Suche nach voll konsistenten Graphen

Wie bereits in Kap. 4.1.1 erläutert wurde, kann es sinnvoll sein, nur nach solchen konsistenten Graphen zu suchen, welche allen Kanten aus dem Eingangsgraphen eine gültige Kantengewichtsbelegung im synthetisierten Graphen zuweisen. Dabei wird das bekannte BT-Verfahren angewendet, für dessen Analyse folgende Voraussetzungen angenommen werden:

- je Kante werden $K_n = K$ Kantengewichte genutzt
- es liegen weniger als K zusammenhängende, voll konsistente Graphen zu Grunde
- die BFS-Suche wird für den aufspannenden Baum verwendet
- die Anzahl konsistenter Kantengewichtskombinationen für die i-te FM ist $\tilde{K}_i = \tilde{K}$ $(1 \leq i \leq N_{\text{FM}})$

Günstigster Fall Im besten Fall entstehen bei der Kombination von zwei FM mit je $\tilde{K}$ konsistenten Kantengesichtskombinationen wieder maximal $\tilde{K}$ konsistente Teil-

graphen. Werden diese mit einer neuen FM mit $\tilde{K}$ konsistenten Kantengewichtskombinationen zusammengeführt, entstehen wieder aus $\tilde{K}^2$ Kombinationen $\tilde{K}$ konsistente Teilgraphen. Dementsprechend sind für jede neue FM gleich viele Kombinationen notwendig, womit sich die Komplexität im besten Fall zu

$$C_{\mathrm{bc}}^{\mathrm{v}} \in \mathcal{O}\left((N-M)\tilde{K}^2\right) \tag{4.9}$$

ergibt, wobei M und N die Knoten- und Kantenanzahl der Komponente sind. Da jede konsistente FM im besten Fall nur drei Kanten hat, können die Vergleichsoperationen der Kantengewichte vernachlässigt werden.

Schlechtester Fall Im schlechtesten Fall lassen sich $\tilde{K}^{N-M+1}$ Kombinationszweige für das BT-Verfahren aus Abb. 4.2 finden, die überprüft werden müssen. Für jeden dieser Kombinationszweige muss der wachsende Graph, initialisiert mit der ersten FM F_1, mit $N-M$ weiteren FM verglichen werden. Dabei werden je weiterem Vergleich maximal M gleiche Kanten der nächsten FM mit dem wachsenden konsistenten Graphen verglichen. Somit ist eine konservative Komplexitätsabschätzung für die Synthese voll konsistenter Graphen

$$C_{\mathrm{wc}}^{\mathrm{v}} \in \mathcal{O}\left((N-M)M\tilde{K}^{(N-M+1)}\right) = \mathcal{O}\left(\tilde{K}^{(N-M+1)}\right). \tag{4.10}$$

Verifikation der Ergebnisse Um die zeitliche Komplexität mit der Laufzeit zu vergleichen, wurde eine C++-Implementierung der Synthese voll konsistenter Graphen auf einem Standard-Desktop-PC mit Linux simuliert. Die Ergebnisse für einen vollständig zusammenhängenden Graphen mit $M = M_{\mathrm{tot}} = 5$ Knoten sind in Abb. 4.8 dargestellt [Kreißig u. Yang, 2012]. Es überwiegt dabei im Fall mit einem BFS-Baum die günstigste Komplexität der Suche nach konsistenten FM C_{cfm}, welche zum Vergleich skaliert eingezeichnet ist. Für den Fall eines DFS-Baumes ist die Komplexität $C_{\mathrm{wc}}^{\mathrm{v}}$ überweigend, wie in Abb. 4.8 auch zu sehen ist.

Neben der Laufzeit der Synthese durch das BT-Verfahren ist in Abb. 4.8 auch die Laufzeit gegeben, wenn alle Kombinationen von konsistenten FM überprüft werden. Dieser Vergleich zeigt den großen Gewinn durch die Synthese. Ebenso ist zum Vergleich die Laufzeit der Synthese mit einem DFS-Baum gegeben. Es ist zu erkennen, dass BFS wie vermutet am effizientesten ist und dass für die Synthese voll konsistenter Graphen die Komplexität hauptsächlich durch die Vorverarbeitung, nämlich die Berechnung der konsistenten FM, bestimmt ist.

4.4.4 Komplexität der CC-Graph-Suche

Um die Effizienzanalyse der CC-Graph-Suche durchführen zu können, führen wir die Knotenanzahl des CC-Graphen $M_{cc} = |V_{cc}| = \sum_{i=1}^{N_{\mathrm{FM}}} \tilde{K}_i$, die maximale Kantenanzahl

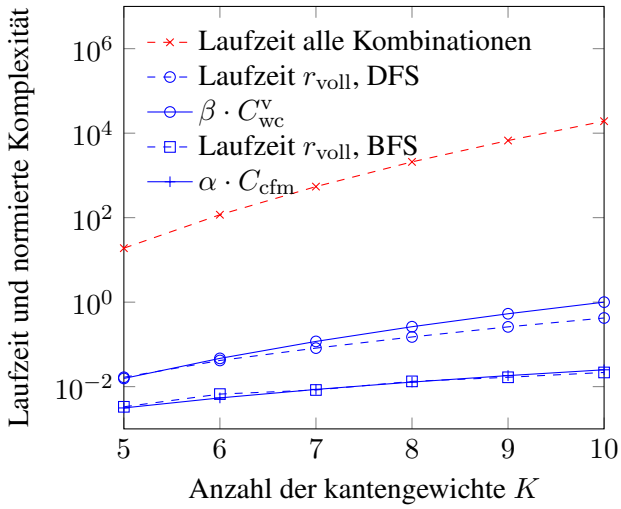


Abbildung 4.8: Laufzeit in Sekunden und Komplexität der Synthese voll konsistenter Graphen mit $\alpha = 5 \cdot 10^{-6}$ und $\beta = 10^{-6}$

$N_{cc} = \binom{M_{cc}}{2}$, die relative Anzahl kompatibler Kanten $p_E = |E(G_{cc})|/N_{cc}$ und die relative Anzahl in Konflikt stehender Kanten $p_{\bar{E}} = |\bar{E}(G_{cc})|/N_{cc}$ ein.

Günstigster Fall Sind alle Knoten im CC-Graph im Konflikt, d.h. $p_{\bar{E}} = 1$, so scheiden sie bereits in der ersten Rekursion von CCGsearch in Alg. 6 aus, weil sie komplett zu $\bar{N}_V(L)$ hinzugefügt werden. Damit ist $N_V(L) = \emptyset$ und jeder Knoten v wird als einzelne Lösung $L = \{v\}$ abgespeichert. Die Anzahl der Aufrufe von CCGsearch ist $B_{\text{bc}} = M_{cc}+1$, da nach dem initialen Aufruf für jeden Knoten im CC-Graphen ein Aufruf erfolgt. Um den jeweiligen Aufruf abzuarbeiten, werden gleich zu Beginn $N_V(L)$ und $\bar{N}_V(L)$ bestimmt, was gemeinsam $M_{cc} - 1$ Operationen benötigt. Anschließend ist CCGsearch abgearbeitet und es erfolgt der Rückgang auf die erste Ebene. Die Komplexität für den effizientesten Fall der Synthese partiell konsistenter Graphen ist somit

$$C^{\text{p}}_{\text{bc}} \in \mathcal{O}\left(M_{cc}(M_{cc} - 1)\right). \tag{4.11}$$

Schlechtester Fall Wenn alle Knoten kompatibel sind, d.h. $p_E = 1$, ist die Reduktion der Knotenmenge V aus Alg. 6 am geringsten, da keine in Konflikt stehenden Nachbarn existieren, d. h. $\bar{N}_V(L) = \emptyset$. Um die Komplexität für diesen Fall zu ermitteln, muss man sich vergegenwärtigen, dass CCGsearch eine vollständige Suche aller Kombinationen durchführt und dass erst nachdem die restlichen Knoten besucht wurden, die doppelten Lösungen nicht gespeichert werden, d. h. alle $\binom{M_{cc}}{i}$ für $3 \leq i \leq M_{cc}$ Knotenkombinationen werden auf jeden Fall durchlaufen. Zu jeder Kombination ist die Komplexität hauptsächlich durch die Operationen bestimmt, um $N_V(L)$ und $\bar{N}_V(L)$ zu berechnen.

Dabei müssen jeweils alle Elemente aus L mit den Elementen aus V verglichen werden. Um deren Kardinalitäten zu erhalten, wird die Rekursionstiefe i betrachtet, die mit dem Hinzufügen eines Knotens zur Lösungsmenge L jeweils um eins steigt. Somit gilt $|L_i| = i$. Demgegenüber sinkt die Anzahl der zu verarbeitenden Knoten $m_i = |V_i|$ um jeweils diesen Knoten. Der Zusammenhang ist direkt durch $m_i = M_{cc} - i$ in der Rekursionstiefe i gegeben. Somit ist zu jedem Funktionsaufruf mit der Länge i der Lösungsmenge die dazugehörige Komplexität $\mathcal{O}\left((M_{cc} - i)i\right)$. Die Gesamtkomplexität für den schlechtesten Fall ist schließlich

$$C_{\mathrm{wc}}^{\mathrm{p}} \in \mathcal{O}\left(\sum_{i=0}^{M_{cc}} \binom{M_{cc}}{i} (M_{cc} - i)i\right) = \mathcal{O}\left(M_{cc}(M_{cc} - 1)2^{M_{cc}-2}\right) = \mathcal{O}\left(M_{cc}^2 \cdot 2^{M_{cc}}\right). \tag{4.12}$$

Um ein Beispiel hierfür zu geben, ist in Abb. 4.9 die Funktionsaufrufkette mit den Mengenbelegungen für einen CC-Graphen mit $M_{cc} = 4$ Knoten gegeben, die alle kompatibel verbunden sind. Nur die markierte Lösung $L = \{v_1, v_2, v_3, v_4\}$ wird gespeichert.

Verifikation der Ergebnisse Ein Vergleich der günstigsten und schlechtesten Komplexität ist in Abb. 4.10 dargestellt. Hierbei wurde eine C++-Implementierung zugrunde gelegt und jeweils ein vollständiger Graph mit $p_E = 1$ oder $p_{\bar{E}} = 1$ simuliert. Die Wiederholungen reichten von 100 bei $M_{cc} = 5$ bis 10 Wiederholungen bei $M_{cc} = 30$. Bei $M_{cc} = 5$ wurde eine Normierung der Komplexität durchgeführt, woraus sich die gezeigten Komplexitäten ergeben: $C_{\mathrm{bc,n}}^{\mathrm{p}} = C_{\mathrm{bc}}^{\mathrm{p}} \cdot \left[r_{\mathrm{bc}}/C_{\mathrm{bc}}^{\mathrm{p}}\right]_{M_{cc}=5}$ und $C_{\mathrm{wc,n}}^{\mathrm{p}} = C_{\mathrm{wc}}^{\mathrm{p}} \cdot \left[r_{\mathrm{wc}}/C_{\mathrm{wc}}^{\mathrm{p}}\right]_{M_{cc}=5}$.

Es ist zu sehen, dass die zwei Grenzfälle einen Unterschied von mehreren Größenordnungen haben. Wird CCGsearch im Rahmen des SONG-Algorithmus eingesetzt, wird es eine Tendenz zu $p_{\bar{E}} = 1$ haben, was an der Generierung der CC-Graph-Knoten liegt, die sich in CCGsearch aus den konsistenten FM berechnen. Diese haben immer gemeinsame Kanten, nämlich die Kanten des aufspannenden Baumes oder die Kanten der gleichen FM bei unterschiedlichen Kantengewichtskombinationen. Diese gemeinsamen Kanten führen stets zu Konflikten, was zu einer Verkürzung der Rechenzeit führt.

Aus den Ergebnissen der akustischen Lokalisierung in Kap. 6.3.5 wissen wir, dass in einem durchschnittlichen Fall weniger als 5% der Knoten kompatibel und ca. 10% im Konflikt sind. Die Laufzeit eines solchen Falls ist auch in Abb. 4.10 eingezeichnet und liegt nahe dem günstigsten Fall. Nachfolgend wird noch der durchschnittliche Fall genauer betrachtet.

Durchschnittlicher Fall Um ein besseres Gefühl für die Komplexität für CCGsearch zu bekommen, wird der durchschnittliche Fall mit variablem p_E und $p_{\bar{E}}$ betrachtet. Mit

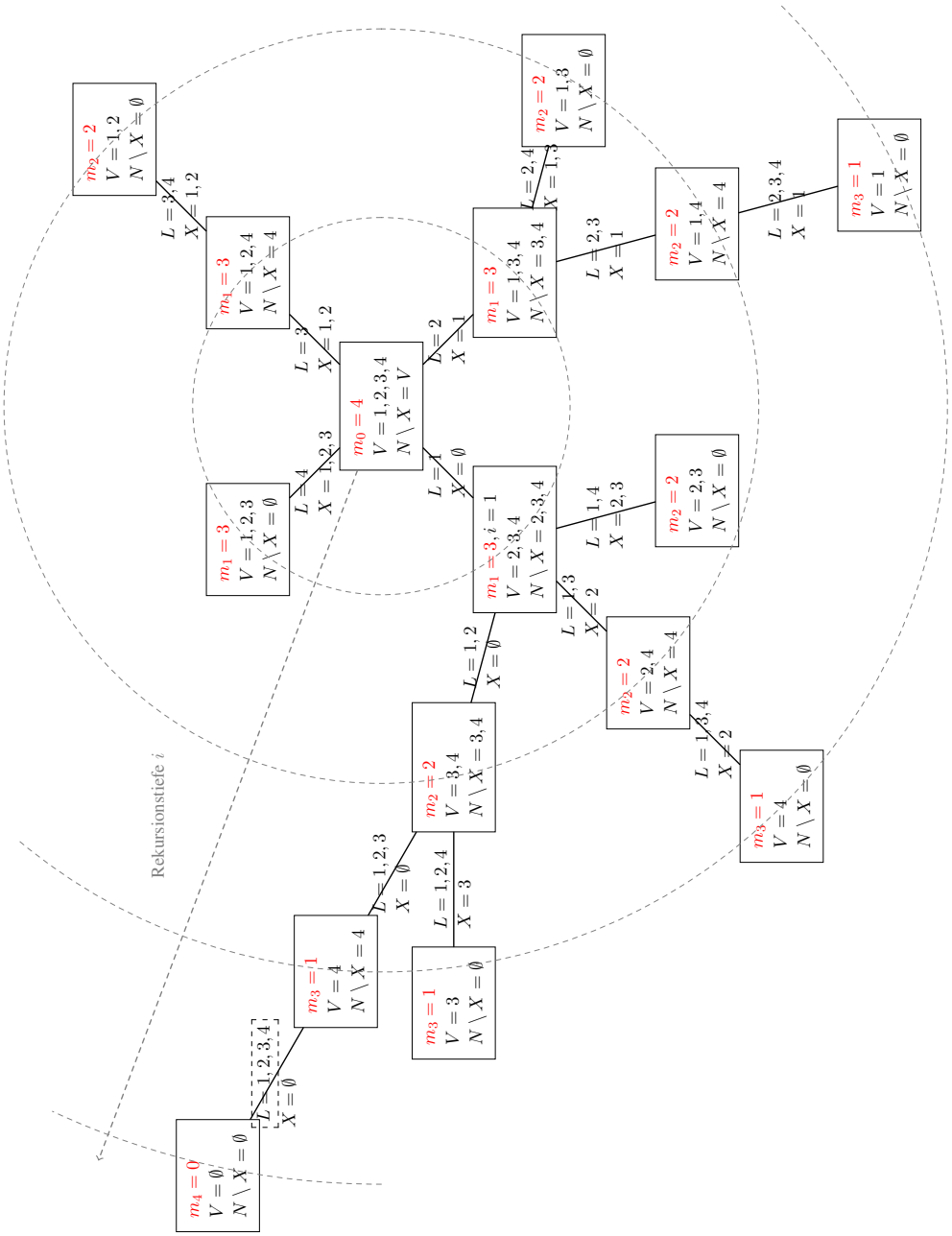


Abbildung 4.9: Schritte des CCGsearch Algorithmus für einen vollständig kompatiblen CC-Graph mit $M_{cc} = 4$ Knoten. Nur die umrandete Lösungsmenge L wird gespeichert.

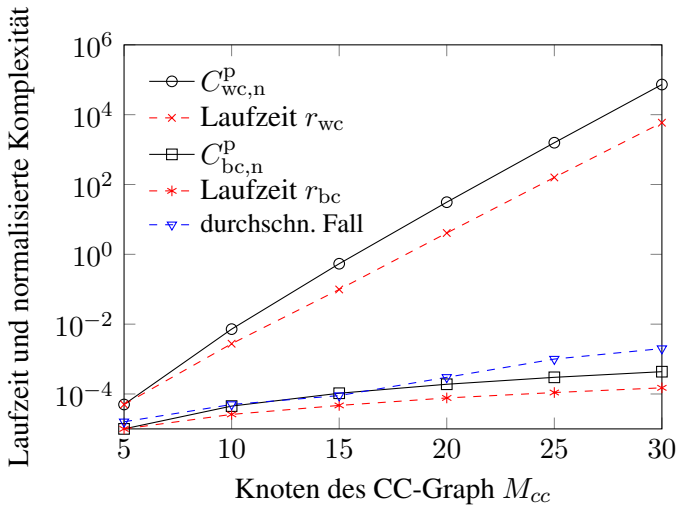


Abbildung 4.10: Vergleich der normierten Komplexitäten (bei $M_{cc} = 5$) im günstigsten und schlechtesten Fall des CCGsearch Algorithmus. Die zugehörigen Laufzeiten und ein durchschnittlicher Anwendungsfall mit $p_E = 5\%$ und $p_{\bar{E}} = 10\%$ sind auch gegeben

m Knoten in einer beliebigen Rekursion ist die mittlere Anzahl kompatibler Nachbarn je Rekursion

$$\mu = \frac{2p_E\binom{m}{2}}{m} = p_E(m-1). \tag{4.13}$$

Dabei ist $p_E\binom{m}{2}$ die durchschnittliche Anzahl kompatibler Kanten mit je 2 Endknoten, die sich auf die m Knoten verteilen. Ebenso ist die mittlere Anzahl in Konflikt stehender Nachbarn $\bar{\mu} = p_{\bar{E}}(m-1)$.

Wie schon für $C^{\mathrm{p}}_{\mathrm{wc}}$ ist die Komplexität für einen Aufruf von CCGsearch $\mathcal{O}(mi)$. Die Kardinalität der Knotenmenge $m = |V|$ wird über das Entfernen der Konfliktnachbarn um $|\bar{N}_V(L)| = \bar{\mu}$ und den aktuell zur Lösungsmenge angefügten Knoten bestimmt. Die Lösungsmenge L steigt nach wie vor immer nur um einen Knoten. Demnach kann eine numerische, rekursive Lösung für die durchschnittliche Komplexität wie folgt angege-

ben werden

$$C_{\mathrm{ac}}(p_E, p_{\bar{E}}) \in \mathcal{O}\left(\sum_{m=0}^{M_{cc}-1} f(M_{cc}-1, 1, m, p_E, p_{\bar{E}})\right) \tag{4.14}$$

$$f(m,i,x,p_E,p_{\bar{E}}) = \begin{cases} m \cdot i + \sum_{\eta=x}^{\lfloor \mu \rfloor} f(m-\bar{\mu}-1, i+1, \eta, p_E, p_{\bar{E}}) & , \ m > 0 \\ 0 & , \ m \leq 0 \end{cases}$$

$$= \begin{cases} m \cdot i + \sum_{\eta=0}^{\lfloor \mu - x \rfloor} f(m-\bar{\mu}-1, i+1, \eta+x, p_E, p_{\bar{E}}) & , \ m > 0 \\ 0 & , \ m \leq 0 \end{cases}.$$

Zuerst werden dabei alle M_{cc} Knoten durchlaufen und in jedem rekursiven Aufruf um $\bar{\mu}+1$ reduziert bis $m=0$ erreicht ist. Dabei repräsentiert x die steigende Kardinalität der bereits besuchten Knoten in X.

Zur Verifikation betrachtet man zunächst den günstigsten Grenzfall $p_E = 0$ und $p_{\bar{E}} = 1$ und erhält aus (4.14) mit $\mu = 0$ und $\bar{\mu} = (m-1)$

$$C_{\mathrm{ac}}(0,1) \in \mathcal{O}\left(\sum_{m=0}^{M_{cc}-1} f(M_{cc}-1, 1, m, 0, 1)\right) \tag{4.15}$$

$$f(m,i,x,0,1) = \begin{cases} m \cdot i + \sum_{\eta=0}^{\lfloor -x \rfloor} f(m-(m-1)-1, i+1, \eta+x, 0, 1) & , \ m > 0 \\ 0 & , \ m \leq 0 \end{cases}. \tag{4.16}$$

Man erkennt, dass die Summe in (4.16) einmalig mit $\eta = 0$ durchlaufen wird und man als Rückgabewert 0 erhält. Somit bekommt man für den ersten Aufruf von (4.16) $(M_{cc}-1)\cdot 1$ und es gilt für (4.15), dass $C_{\mathrm{ac}}(0,1) = \sum_{0}^{(M_{cc}-1)}(M_{cc}-1) = M_{cc}(M_{cc}-1) = C_{\mathrm{bc}}^{\mathrm{p}}$.

Der andere Grenzfall der schlechtesten Komplexität mit $p_E = 1$ und $p_{\bar{E}} = 0$ ist beschrieben durch $\mu = (m-1)$ und $\bar{\mu} = 0$. Somit ist

$$C_{\mathrm{ac}}(1,0) \in \mathcal{O}\left(\sum_{m=0}^{M_{cc}-1} f(M_{cc}-1, 1, m, 1, 0)\right) \tag{4.17}$$

$$f(m,i,x,1,0) = \begin{cases} m \cdot i + \sum_{\eta=0}^{\lfloor m-1-x \rfloor} f(m-1, i+1, \eta+x, 1, 0) & , \ m > 0 \\ 0 & , \ m \leq 0 \end{cases}.$$

Da (4.17) in rekursiver Form gegeben ist, zeigt Abb. 4.11 einen Vergleich von $C_{\mathrm{wc}}^{\mathrm{p}}$ aus (4.12) und der numerischen Auswertung von $C_{\mathrm{ac}}(1,0)$ aus (4.17). Es ist zu sehen, dass beide Funktionen die gleichen Ergebnisse liefern. Ebenso ist auch ein Vergleich von $C_{\mathrm{ac}}(0,1)$ aus (4.15) und $C_{\mathrm{bc}}^{\mathrm{p}}$ von (4.11) gegeben.

Um die Ergebnisse zur durchschnittlichen Komplexität weiter zu verifizieren, wurde ein Vergleich mit der Laufzeit durchgeführt. Dabei wurden CC-Graphen generiert, die

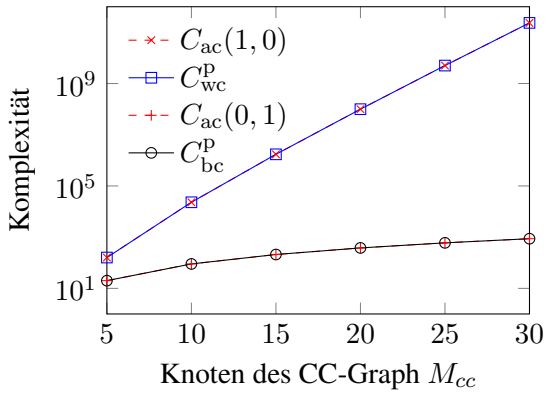
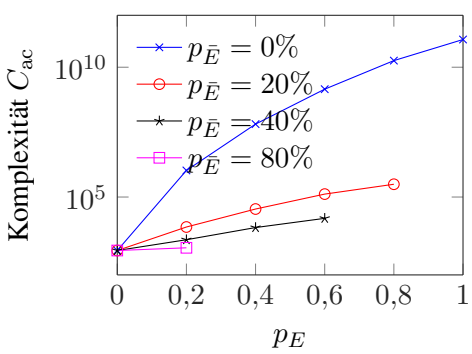


Abbildung 4.11: Numerische Lösung von (4.17) und analytische Lösung der Komplexität nach (4.12) und (4.11)

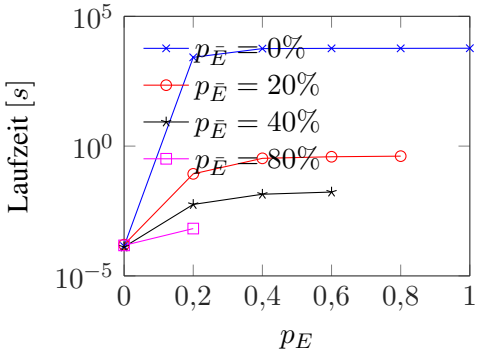
entsprechend der Vorgabe $(p_E, p_{\bar{E}})$ kompatible und in Konflikt stehende Knotenpaare aufweisen. Bei der Generierung gilt immer, dass $p_E + p_{\bar{E}} \leq 1$. Die Laufzeit wurde aus der C++-Implementierung bestimmt und über 20 – 1 000 CC-Graphen je $(p_E, p_{\bar{E}})$-Paar gemittelt.

In Abb. 4.12 sind die Ergebnisse aus (4.14) und die Laufzeit dargestellt. Für geringe Werte von $p_{\bar{E}}$ sieht man in Abb. 4.12(a) und (b) den rapiden Anstieg sowohl der Komplexität als auch der Laufzeit mit steigendem p_E. Für große Werte von $p_{\bar{E}}$ erkennt man dort auch, dass eine Variation von p_E geringere Auswirkungen hat. Das ist zu erwarten, denn die relative Anzahl der Konfliktkanten $p_{\bar{E}}$ bewirkt eine Verkleinerung der Knotenmenge V_{cc} in jeder Rekursion von CCGsearch. Je höher $p_{\bar{E}}$, desto stärker diese Reduktion. Für geringe Werte von $p_{\bar{E}}$ ist der Einfluss der kompatiblen Kanten p_E größer, da weniger Knoten je Rekursion entfernt und mehr mögliche Lösungsmengen verfolgt werden, die aber schlussendlich auf die gleichen Lösungen hinzielen.

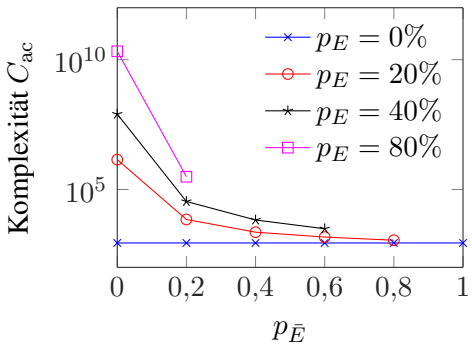
Diese Abhängigkeit ist auch in Abb. 4.12(c) und (d) zu erkennen, in denen für feste p_E der Abstieg der Komplexität und Laufzeit für steigende $p_{\bar{E}}$ gezeigt ist. Je größer die relative Anzahl der Konfliktkanten $p_{\bar{E}}$ umso geringer ist die Laufzeit, unabhängig von der relativen Anzahl kompatibler Kanten. Dementsprechend ist für eine geringe Laufzeit von CCGsearch eine hohe Anzahl von inkompatiblen Knoten notwendig. Interessant ist in Abb. 4.12(c) und (d) auch die nahezu konstante Laufzeit bei $p_E = 0\%$. Diese stammt von den fehlenden kompatiblen Nachbarn, weshalb CCGsearch keinen einzigen Rekursionsaufruf durchführt.



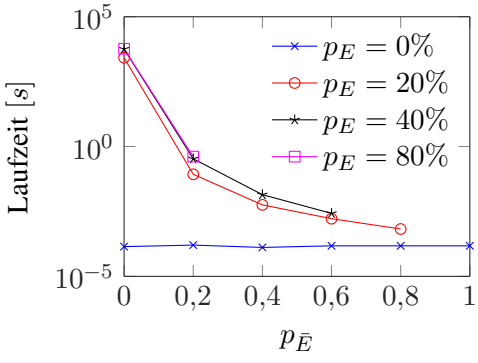
(a) Komplexität C_{ac} in Abhängigkeit von der relativen Anzahl kompatibler Kanten p_E für unterschiedliche $p_{\bar{E}}$



(b) Laufzeit von CCGsearch in Abhängigkeit von der relativen Anzahl kompatibler Kanten p_E für unterschiedliche $p_{\bar{E}}$



(c) Komplexität C_{ac} in Abhängigkeit von der relativen Anzahl Konfliktkanten $p_{\bar{E}}$ für unterschiedliche p_E



(d) Laufzeit von CCGsearch in Abhängigkeit von der relativen Anzahl Konfliktkanten $p_{\bar{E}}$ für unterschiedliche p_E

Abbildung 4.12: Vergleich der durchschnittlichen Komplexität C_{ac} des CCGsearch-Algorithmus mit den gemessenen Laufzeiten aus Simulationen für CC-Graphen mit $M_{cc} = 30$ Knoten

4.4.5 Komplexität von SONG

Der SONG-Algorithmus, wie er in Kap. 4.3 erläutert wurde, ist in Tab. 4.4 mit seinen einzelnen Komplexitäten aufgelistet.

Es ist zu erkennen, dass die Wahl der Anzahl der Kantengewichte K und die Anzahl der konsistenten FM $\tilde{K}$ einen großen Einfluss auf die Komplexität haben. Außerdem können die Vorverarbeitungsschritte wie die Komponentensuche, die Suche nach dem aufspannenden Baum und die FM-Berechnung nahezu vernachlässigt werden, sofern $K^3 > M$ gilt.

Verarbeitungsschritt	Komplexität bester Fall	schlechtester Fall
Komponentensuche	$C_{\mathrm{bi}} = \mathcal{O}\left(M_{\mathrm{tot}} + N_{\mathrm{tot}}\right)$	
aufspannender Baum	$C_{\mathrm{st}} = \mathcal{O}\left(M + N\right)$	
fundamentale Maschen	$C_{\mathrm{fm}} = \mathcal{O}\left((N - M + 1)M\right)$	
konsistente FM	$C_{\mathrm{cfm}}^{\mathrm{bc}} = \mathcal{O}\left((N - M + 1)K^3\right)$	$C_{\mathrm{cfm}}^{\mathrm{wc}}\mathcal{O}\left((N - M + 1)K^M\right)$
Backtracking:	$C_{\mathrm{bc}}^{\mathrm{v}} \in \mathcal{O}\left((N - M + 1)\tilde{K}^2\right)$	$C_{\mathrm{wc}}^{\mathrm{v}} \in \mathcal{O}\left(\tilde{K}^{N-M+1}\right)$
CCGsearch	$C_{\mathrm{bc}}^{\mathrm{p}} \in \mathcal{O}\left(M_{cc}^2\right)$	$C_{\mathrm{wc}}^{\mathrm{p}} \in \mathcal{O}\left(M_{cc}^2 \cdot 2^{M_{cc}}\right)$

Tabelle 4.4: Zusammenfassung der einzelnen Komplexitäten des SONG-Algorithmus

Gewöhnlich wird K größer als die gewünschte Anzahl konsistenter Graphen, wie die Anzahl der Quellen in der Lokalisierung, gewählt, um keine Lösung zu verlieren. Um den größten Aufwand zu bestimmen, setzen wir C_{cfm} zunächst mit $C_{\mathrm{bc}}^{\mathrm{v}}$ und danach mit $C_{\mathrm{bc}}^{\mathrm{p}}$ ins Verhältnis. Somit lässt sich für die Synthese voll konsistenter Graphen folgende Aussage treffen: Ist $K^3 > \tilde{K}^2$, überwiegt die Suche nach konsistenten Kombinationen die Synthese mittels BT. Für die partielle Synthese konsistenter Graphen ist $M_{cc} = (N - M + 1)\tilde{K}$ und dann überwiegt CCGsearch bei $(N - M + 1)\tilde{K}^2 > K^3$. Es ist gut zu erkennen, dass im besten Anwendungsfall die volle und die partielle Synthese sich nur um den Faktor $N - M + 1$ in ihrer Komplexität unterscheiden.

4.5 Simulationsergebnisse der zeitlichen Abhängigkeiten

Da die vorherigen Betrachtungen die einzelnen Komponenten im SONG-Algorithmus auf ihre besten und schlechtesten Komplexitäten getestet haben, betrachtet dieses Kapitel das Laufzeitverhalten des gesamten Algorithmus. Dafür werden unterschiedliche Potenzialwerte je Knoten generiert, wie sie die Signallaufzeiten $t_{iq,0}$ von einer Quelle q zu jedem Mikrofon i bei der Lokalisierung darstellen. Daraus lassen sich die konsistenten Kantengewichte als Differenz der jeweiligen Potenziale berechnen (gleichbedeutend der TDOA für die Lokalisierung). Ist ein Graph mit $N < N_{\max}$ für eine Simulation gesucht, dann entfernen wir nur solche Kanten, dass der resultierende Graph zweifachzusammenhängend ist. Somit gilt für dieses Kapitel $M = M_{\mathrm{tot}}$ und $N = N_{\mathrm{tot}}$.

Die zufällig generierten Graphen wurden anschließend mit dem SONG-Algorithmus verarbeitet, wobei ein BFS-Baum mit einem Startknoten mit maximalem Knotengrad verwendet wurde. Die Laufzeit wurde mit der ctime-Bibliothek für die C++-Implementierung gemessen. Es wurde ein Standard AMD-Dual-Core-PC verwendet, der mit einer Ubuntu Linux Distribution der Version 12.04 ausgestattet war.

4.5.1 Von der Knotenanzahl bei konstantem N_{FM}

Um die Abhängigkeit der Laufzeit von der Knotenanzahl zu bestimmen, variiert man nicht nur die Knotenanzahl M, sondern mit ihr die Anzahl der konsistenten Maschen N_{FM}. Genauer bedeutet dies, dass die Kantenanzahl so geändert wird, dass $N-M+1 =$ konst. gilt. Damit kann das Anwachsen des Graphen mit zunehmender Knotenanzahl und somit den Einfluss der Kantenanzahl kompensiert werden. Dieser Zusammenhang wird auch aus der Komplexität in Tab. 4.4 ersichtlich, da die Komplexitäten hauptsächlich mit $(N - M + 1)$ skalieren.

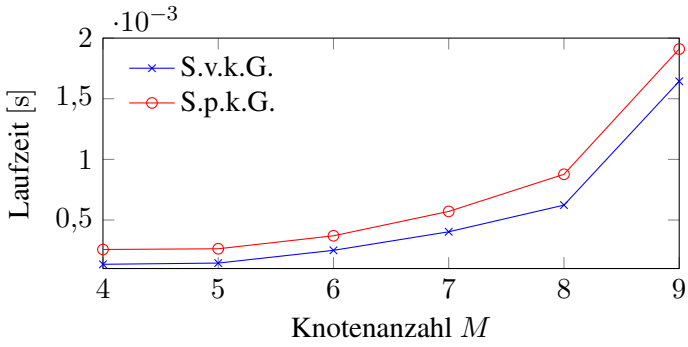


Abbildung 4.13: Anstieg der Laufzeit von SONG mit steigender Knotenanzahl bei $N_{\mathrm{FM}} = 3$ und $K = 4$ (S.v.k.G.: Synthese voll konsistenter Graphen, S.p.k.G.: Synthese partiell konsistenter Graphen)

Da der kleinste Graph $M = 4$ Knoten besitzt, ist $N - M + 1 = 3$ die minimal mögliche Konstante. Es wurden je Knotenanzahl $S = 100$ unterschiedliche Graphen generiert mit jeweils $K = 4$ konsistenten Kantengewichtsbelegungen. In Abb. 4.13 sind die Laufzeiten der Synthese voll und partiell konsistenter Graphen zu sehen, die über 100 Wiederholungen gemittelt wurden. Es ist gut der multiplikative Unterschied von $C_{\mathrm{bc}}^{\mathrm{p}}/C_{\mathrm{bc}}^{\mathrm{v}} = N - M + 1$ erkennbar. Trotz des quadratischen Anstiegs der Rechenzeit mit wachsendem M sind bei konstantem N_{FM} beide Verfahren für einen Standard-PC nicht zeitkritisch.

4.5.2 Von der Kantenanzahl bei fester Knotenanzahl

Da in Abb. 4.13 die Kantenanzahl durch die konstante Anzahl der FM $N_{\mathrm{FM}} = 3$ proportional mit der Knotenanzahl wächst, nämlich $N = M + 2$, betrachten wir in diesem Abschnitt die Abhängigkeit der Laufzeit von der wachsenden Kantenanzahl bei einem Graphen mit $M = 8$ Knoten. Die maximale Kantenanzahl des vollständig zusammenhängenden Graphen ist dann $N_{\max} = 28$. Im Vergleich zum konstanten N_{FM}, wirkt sich die Anzahl der Kanten stärker auf die Laufzeit aus, wie Abb. 4.14 zeigt.

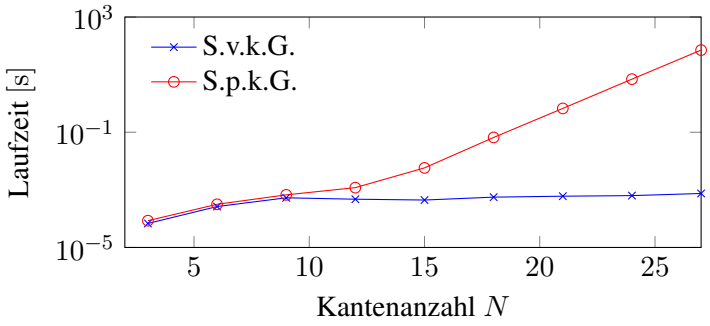


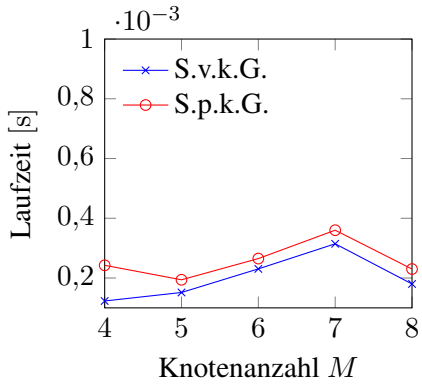
Abbildung 4.14: Laufzeit von SONG in Abhängigkeit der Kantenanzahl für zweifach-zusammenhängende Graphen mit $M = 8$ und $K = 4$ (S.v.k.G.: Synthese voll konsistenter Graphen, S.p.k.G.: Synthese partiell konsistenter Graphen)

Es zeigt sich, dass die Synthese partiell konsistenter Graphen einen exponentiellen Anstieg erfährt, während die Synthese voll konsistenter Graphen ab $N = 9$ nur polynomiell steigt. Das deutet darauf hin, dass bis $N = 9$ die Vorverarbeitung überwiegt und erst danach das Syntheseverfahren den Hauptanteil der Berechnung ausmacht.

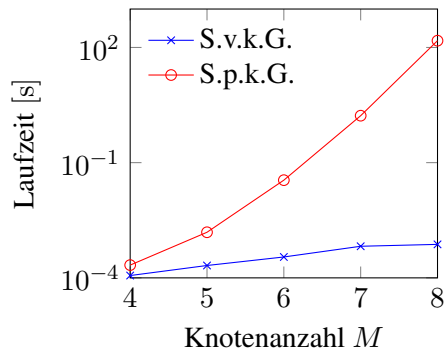
4.5.3 Von der Knotenanzahl bei maximaler Kantenanzahl

Um die Auswirkungen der Knotenanzahl zu untersuchen, wurde zunächst eine feste Kantenanzahl $N = 6$ gewählt und die Knotenanzahl variiert. Wie in Abb. 4.15(a) zu sehen ist, hat die Änderung der Knotenanzahl nur einen geringen Anteil an der Gesamtlaufzeit von SONG. Die Variation in der Laufzeit ist marginal und hauptsächlich in der Topologie der generierten Graphen begründet, insbesondere in der Verteilung der $N = 6$ zweifach-zusammenhängenden Kanten. Diese Topologie hat direkte Auswirkungen auf die Länge der FM.

Da die Simulation mit festem N und variablem M aufgrund der geringen maximalen Kantenanzahl $N_{\max} = 6$ für $M = 4$ und der Vergleich zu $N_{\max} = 28$ bei $M = 8$ schwierig ist, wurde zusätzlich eine Simulation mit steigender maximaler Kantenanzahl durchgeführt. Es ist in Abb. 4.15(b) zu erkennen, dass die volle Synthese wie bereits in Abb. 4.14 einen polynomiellen Anstieg mit der Knoten- bzw. Kantenanzahl $N_{\max} = \binom{M}{2}$ zeigt. Für die Laufzeit der Synthese partiell konsistenter Graphen sehen wir, dass die Laufzeit über die Knotenanzahl ebenso exponentiell steigt wie in Abb. 4.14. Man kann somit zusammenfassen, dass topologieseitig besonders die Kantenanzahl einen großen Einfluss auf die Laufzeit hat.



(a) Laufzeit mit $N = 6$ über die Knotenanzahl

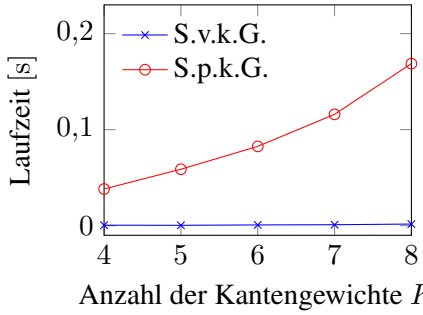


(b) Laufzeit bei vollständig zusammenhängenden Graphen mit $N = \binom{M}{2}$

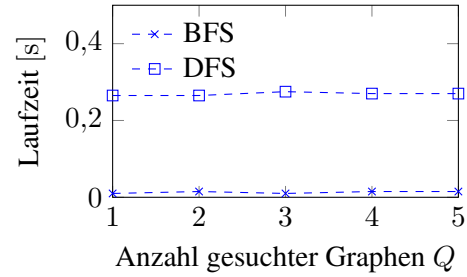
Abbildung 4.15: Laufzeit des SONG-Algorithmus über die Knotenanzahl mit $K = 4$ (S.v.k.G.: Synthese voll konsistenter Graphen, S.p.k.G.: Synthese partiell konsistenter Graphen)

4.5.4 Von der Kantengewichtsanzahl

Der letzte wichtige Parameter für die Synthese ist die Anzahl der Kantengewichte K. Wie in Abb. 4.16(a) zu sehen ist, steigt die Laufzeit entsprechend Tab. 4.4 polynomiell für einen vollständig zusammenhängenden Graphen mit $M = 6$ Knoten und $N = 15$ Kanten. Die Anzahl der Kantengewichte K entsprechen hierbei direkt der Anzahl gewünschter voll konsistenter Graphen Q.



(a) Anstieg der Laufzeit für einen vollständig zusammenhängenden Graphen mit $M = 6$ Knoten, in Abhängigkeit von der Anzahl der Kantengewichte K



(b) Laufzeit der S.v.k.G. für unterschiedliche gewünschte konsistente Graphen Q bei fester Kantengewichtsanzahl $K = 10$ ($M = 5, N = 10$)

Abbildung 4.16: Laufzeiten von SONG für eine unterschiedliche Anzahl von Kantengewichten K und darin unterschiedliche voll konsistente Graphen Q

Stammen die Kantengewichte von zufälligen anderen Werten und nicht von den gesuchten konsistenten Graphen, ist dies für die Gesamtkomplexität vernachlässigbar, wie Abb. 4.16(b) zeigt. Hierbei wurden stets $K = 10$ Kantengewichte generiert, wobei $Q = [1, \ldots, 5]$ gesuchte Werte zu voll konsistenten Graphen gehören. Es zeigt sich, dass die Rechenzeit für die Synthese voll konsistenter Graphen eine nahezu konstante Rechenzeit benötigt. Weiterhin erkennt man, dass der zusätzliche Rechenaufwand aus den langen FM eines DFS-Baumes wie erwartet eine größere Rechenzeit benötigt.

Aus diesen Beobachtungen lässt sich schlussfolgern, dass für die volle Synthese die gesamte Kantengewichtsanzahl K und somit die Komplexität C_{cfm} entscheidend ist. Für die Synthese partiell konsistenter Graphen überwiegt nach wie vor CCGsearch, welcher durch $M_{cc} \sim \tilde{K}$ stark von den konsistenten Gewichtskombinationen abhängt.

Kapitel 5

Analyse der Verfahren zur Synthese konsistenter Graphen

Für die Synthese konsistenter Graphen ist die Topologie ein entscheidender Faktor. Wird ein zusammenhängender Eingangsgraph mit $N \leq \binom{M}{2}$ Kanten übergeben, so sucht SONG einen BFS- oder DFS-Baum in allen Kanten. Sind in diesem Eingangsgraphen fehlerhafte Kantengewichtsmengen vorhanden, so kann es zu fehlenden konsistenten Graphen kommen. In diesem Kapitel werden die Auswirkungen von fehlenden Kantengewichten analysiert, wenn nur $N_\theta \leq N$ Kantengewichte gültig sind, aber N Kantengewichte übergeben werden. Hierfür werden konsistente, zweifach-zusammenhängende Graphen mit N_θ Kanten mittels des Verfahrens aus Kap. 4.5 generiert und anschließend werden die fehlenden Kantengewichte mit zufälligen Werten aufgefüllt. Somit entsteht ein vollständig zusammenhängender Graph, der $N_{\max} - N_\theta$ falsche Kantengewichte enthält. Die Detektionsrate P_D ist die Anzahl, wie oft der generierte, konsistente Teilgraph mit N_θ gültigen Kantengewichten gefunden wurde, geteilt durch die Anzahl der Simulationsdurchläufe S, wobei je Simulation zunächst nur ein Graph generiert wird, $P_D \leq 1$. Die Falschalarmrate P_F ist die Anzahl der zusätzlich synthetisierten, konsistenten Graphen, geteilt durch die Anzahl der Simulationsdurchläufe S, $P_F \geq 0$.

5.1 Detektionsrate der Synthese

In Abb. 5.1 sind sowohl die Detektions- als auch die Falschalarmrate für die Synthese partiell und voll konsistenter Graphen gezeigt. Dabei wurden $S = 1000$ Graphen mit $K = 1$ Kantengewicht und $M = 6$ Knoten generiert. Für die Synthese wurden $N = N_{\max}$ Kanten übergeben, wobei nur N_θ Kantengewichte zu einem konsistenten, zweifach-zusammenhängenden Teilgraphen gehörten.

Wie erwartet findet die Synthese partiell konsistenter Graphen teilweise die konsistenten Teilgraphen, wobei die Detektionsrate mit abnehmendem N_θ sinkt. Im Gegensatz dazu findet die Synthese voll konsistenter Graphen nur die Lösungen mit allen $N_\theta = N$ Kantengewichten.

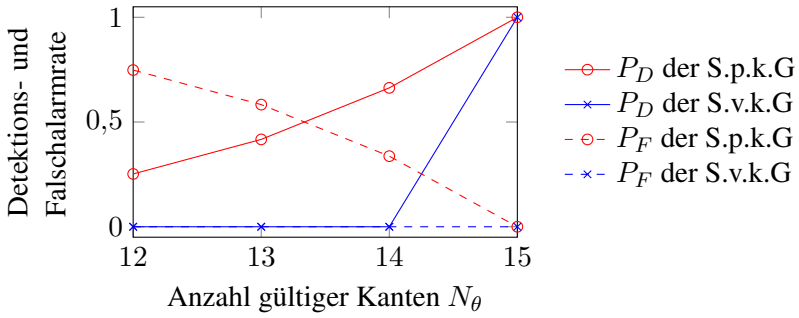


Abbildung 5.1: Erkennung der Synthese voll und partiell konsistenter Graphen (S.v.k.G, S.p.k.G.) mit unterschiedlich gültigen Kantengewichtsmengen

Die Falschalarmrate zeigt, dass die Synthese voll konsistenter Graphen keine zusätzlichen Lösungen findet und somit sehr robust ist. Demgegenüber findet die Synthese partiell konsistenter Graphen für einige Fälle bei $N_\theta < N$ zusätzliche konsistente Teilgraphen, die allerdings mit zunehmendem N_θ weniger werden.

5.2 Detektionsraten der unterschiedlichen aufspannenden Bäume

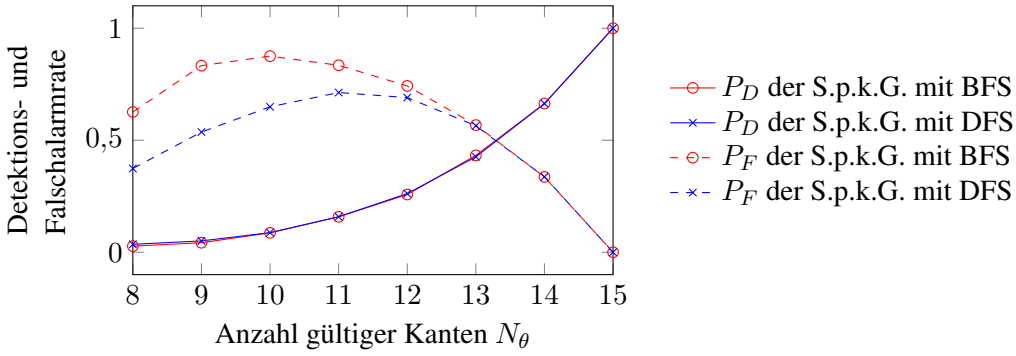


Abbildung 5.2: Detektions- und Falschalarmrate mit jeweils BFS oder DFS aufspannendem Baum und partieller Synthese

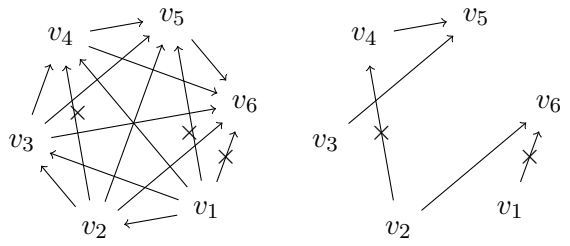
An dieser Stelle werden noch einmal die Auswirkungen der unterschiedlichen aufspannenden Bäume genauer betrachtet. Hierzu wurden die gleichen, partiell konsistenten Graphen aus Abb. 5.1 mittels des BFS- und DFS-Verfahrens synthetisiert. In Abb. 5.2 sieht man die zugehörigen Detektions- und Falschalarmraten aus der Synthese partiell konsistenter Graphen. Beide aufspannenden Bäume finden die jeweiligen gesuchten,

partiell konsistenten Graphen gleich gut. Es ist lediglich ein Unterschied in der Falschalarmrate zu beobachten. Sie zeigt, dass die Syntheseverfahren mit einem BFS-Baum mehr zusätzliche, konsistente Lösungen finden. Eine Erklärung hierfür ist die Länge der FM, die bei einem BFS-Baum kürzer sind als bei einem DFS-Baum. Dementsprechend kann es bei vielen ungültigen Kantengewichten bzw. kleinem N_θ, eher dazu kommen, dass sich bei einer kurzen FM zufällig eine Nullsumme ergibt, woraus ein konsistenter Teilgraph resultiert. Bei den langen FM des DFS-Baumes müssen hingegen mehrere Kantengewichte passen, damit zufällig eine konsistente Lösung gefunden wird.

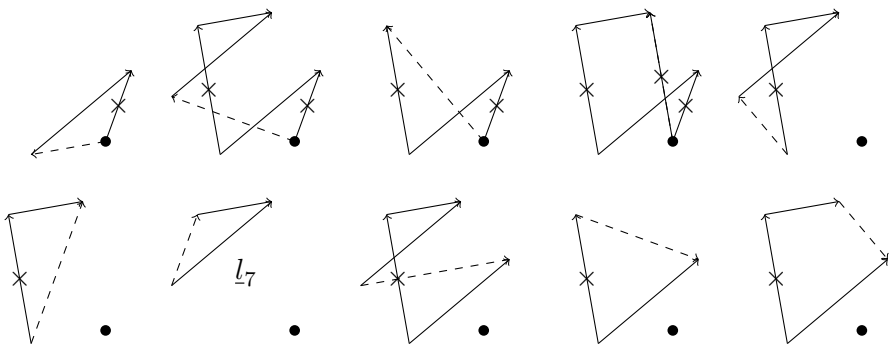
5.3 Auswirkungen der fehlenden Kantengewichte

5.3.1 Grafische Analyse

Trotz der partiellen Konsistenz kann es in ungünstigen Konstellationen dazu kommen, dass nicht der gewünschte Teilgraph mit $N_\theta < N$ Kanten gefunden wird, sondern nur ein kleinerer Teilgraph, wie die Detektionsraten in den vorherigen Kapitel zeigen.



(a) Ursprungsgraph mit den ungültigen Kanten (b) Aufspannender Baum



(c) Zugehörige FM

Abbildung 5.3: Auswirkungen eines schlecht gewählten aufspannenden Baumes

Dieses Phänomen wird in Abb. 5.3 gezeigt, die einen möglichen Eingangsgraphen in Abb. 5.3(a) mit $M = 6$ Knoten und $N_\theta = 12$ konsistenten Kantengewichten statt der maximal möglichen 15 Kanten zeigt. Die restlichen drei mit $\times$ markierten Kanten stellen fehlerhafte Messungen dar, die mit zufälligen Werten übergeben wurden und somit in die Topologie der Synthese mit einfließen. Dadurch fehlen dem zu synthetisierenden Graphen nur drei Werte.

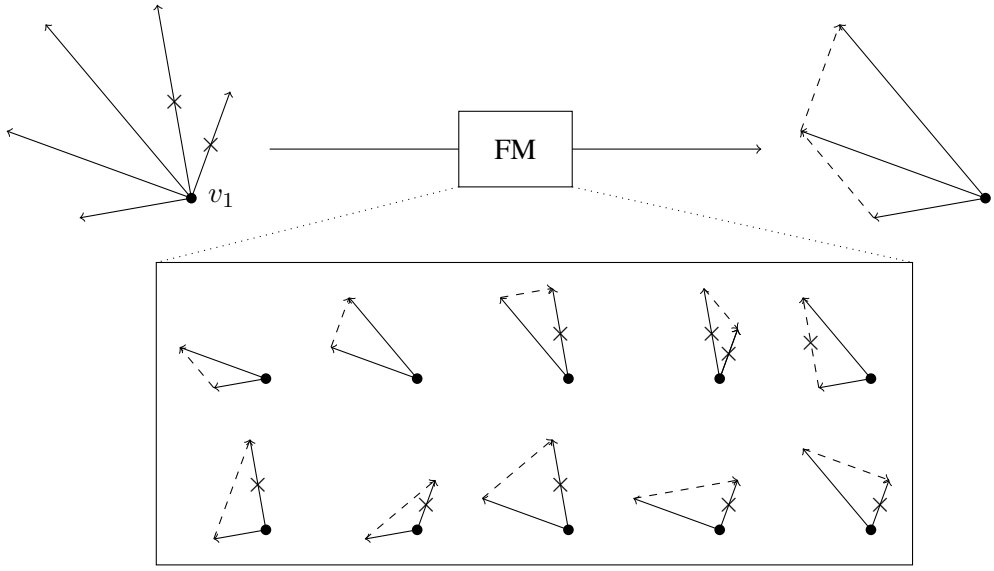
In der Simulation wurde nun ein DFS-Baum generiert, der in Abb. 5.3(b) dargestellt ist und mit dem Wurzelknoten v_1 initialisiert wurde. Wie man schon in Abb. 5.3(b) erkennt, sind zwei der drei ungültigen Kanten im aufspannenden Baum enthalten, was sich stark auf die resultierenden FM in Abb. 5.3(c) auswirkt. Bei genauerer Betrachtung erkennt man, dass nur die FM $\underline{l}_7$ in Abb. 5.3(c) eine konsistente Lösung finden kann und damit nur ein konsistenter Graph mit drei gültigen Kantengewichten gefunden werden kann. Wählt man hingegen einen BFS-Baum mit dem Wurzelknoten v_1, so stellen sich die FM wie in Abb. 5.4(a) dar. Es sind immer noch viele FM nicht synthetisierbar. Allerdings ist der resultierende Graph größer als beim DFS-Baum aus Abb. 5.3. Es ist in Abb. 5.4(a) zu erkennen, dass die ungültigen Kanten (v_1, v_5) und (v_1, v_6) im aufspannenden Baum mehrere FM stören und die ungültige Kante aus dem Kobaum nur eine FM betrifft. Dies ist eine logische Konsequenz, da jede Kobaumkante nur einmal in der FM-Matrix $\mathbf{L}_{\mathrm{FM}}$ vorkommt. Die beste Wahl für den vorliegenden Graphen ist somit ein BFS-Baum mit dem Wurzelknoten v_3, wie er in Abb. 5.4(b) gezeigt ist, da alle gestörten Kanten eine Kobaumkante sind.

Dieses Beispiel zeigt den starken Zusammenhang zwischen dem Wurzelknoten, dem aufspannenden Baum und der relativen Lage der ungültigen Kanten zueinander. Da für SONG bisher keine weiteren A-priori-Informationen berücksichtigt wurden, ist eine weitere Verbesserung nicht möglich.

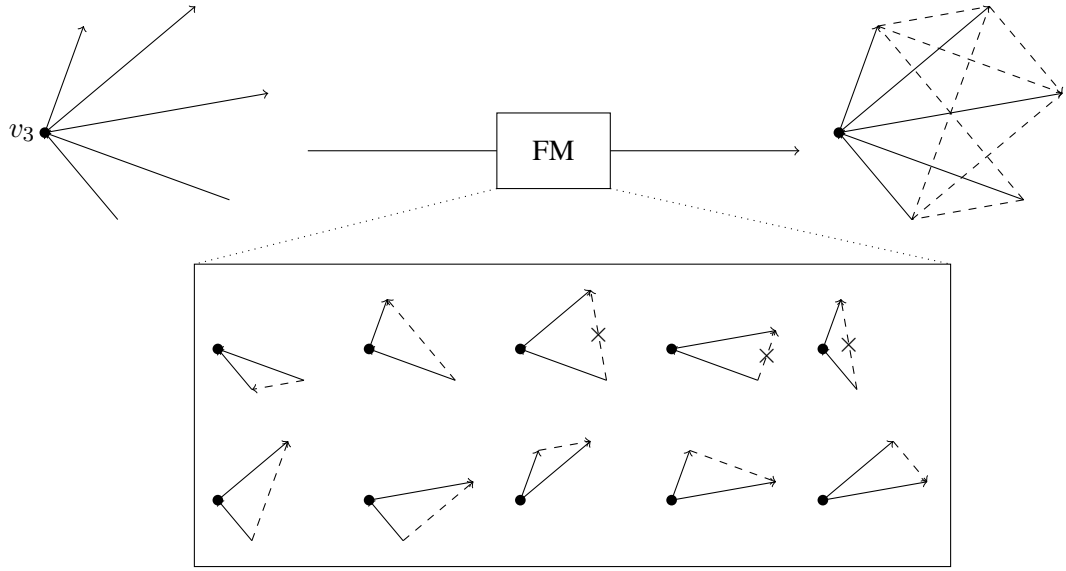
5.3.2 Mathematische Analyse

Wie bereits diskutiert muss unterschieden werden, ob die gestörte Kante im aufspannenden Baum oder im Kobaum liegt. Liegt sie im Kobaum, so wird nur die FM gestört, welche von der jeweiligen Kante geschlossen wird. Dieser Fall ist in der Synthese partiell konsistenter Graphen weniger kritisch wie Abb. 5.4(b) zeigt. Befindet sich die gestörte Kante im aufspannenden Baum, so wirkt sie sich unter Umständen auf mehrere FM aus, wie das Beispiel in Abb. 5.3 und Abb. 5.4(a) veranschaulicht. Wie viele FM gestört werden, kann gut aus der zugehörigen Spalte der FM-Matrix $\mathbf{L}_{\mathrm{FM}}$ abgelesen werden.

Im Folgenden werden die Auswirkungen einer ungültigen Kanten im BFS- und DFS-Baum mathematisch analysiert. Mit konstantem M, zunehmender Kantenanzahl N und gleichbleibendem N_θ steigt auch die Anzahl der FM N_{FM} und somit die Möglichkeit



(a) BFS-Baum mit dem Wurzelknoten v_1 aus Abb. 5.3(a)



(b) BFS-Baum mit dem Wurzelknoten v_3 aus Abb. 5.3(a)

Abbildung 5.4: Auswirkungen des Wurzelknotens auf einen BFS-Baum

für weitreichende Störungen. Deshalb ist der Grenzfall einer maximalen Beeinflussung für vollständig zusammenhängende Graphen gegeben. In diesem Fall ist der aufspannende BFS-Baum ein Sterngraph mit dem Wurzelknoten im Mittelpunkt und die FM haben alle die Länge drei. Daraus kann man ableiten, dass eine gestörte Kante des BFS-Baumes genau $l_{\mathrm{BFS}} = M - 2$ FM beeinflusst, die mit den Endknoten der gestörten

Kante und den übrigen $M-2$ Knoten die FM der Länge drei bilden. Demnach sind $(N-M+1)-(M-2)=\frac{1}{2}(M-2)(M-3)$ FM ungestört.

Für einen DFS-Baum in einem vollständig zusammenhängenden Graphen gilt, dass er einen Pfad der Länge $M-1$ und eine FM mit maximaler Länge M besitzt, wie die zweite FM in Abb. 5.3(c). Um die Anzahl gestörter FM zu ermitteln, wird zunächst das Beispiel in Abb. 5.5 betrachtet.

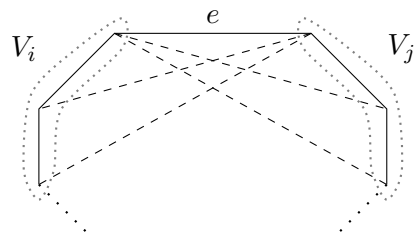


Abbildung 5.5: Gestörte Kante e eines DFS-Baumes (durchgezogene Linien) und deren Auswirkung auf die FM (geschlossen durch die gestrichelten Linien des Kobaumes)

Hier ist die Kante e gestört und teilt die Knotenmenge in die Mengen V_i und V_j. Es ist zu erkennen, dass e genau die FM stört, die durch alle Kombinationen zwischen V_i und V_j erzeugt werden. Angenommen $m=|V_i|$ $(1\leq m\leq M-1)$ und damit $M-m=|V_j|$, dann ist die Anzahl der Knotenpaare zwischen den zwei Mengen gegeben durch $m(M-m)$. Da e auch zu diesen Kombinationen gehört, ist die Anzahl gestörter FM $l_{\mathrm{DFS}}=m(M-m)-1$. Die Anzahl der ungestörten FM somit $N-M+1-l_{\mathrm{DFS}}$. Die Auswirkungen des DFS-Baumes auf die FM sind in Tab. 5.1 in Abhängigkeit von der Knotenanzahl dargestellt.

M	l_{DFS} maximal bei	Anzahl gestörter FM	Anzahl ungestörter FM
gerade	$m=M/2$	$l_{\mathrm{DFS}}=\left(\frac{M}{2}\right)^2-1$	$\frac{1}{4}(M-2)(M-4)$
ungerade	$m=(M\pm1)/2$	$l_{\mathrm{DFS}}=\left(\frac{M+1}{2}\right)\left(\frac{M-1}{2}\right)-1$	$\frac{1}{4}(M-3)^2$

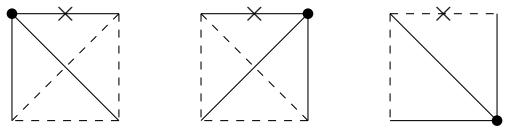
Tabelle 5.1: Anzahl gestörter FM für einen DFS Baum

Für den Grenzfall $m=1$ bzw. $m=M-1$ ist $l_{\mathrm{DFS}}=l_{\mathrm{BFS}}=M-2$, was den besten Fall darstellt. Somit ist die Wahl eines BFS-Baumes besser im Bezug auf die Anfälligkeit für fehlende konsistente Messwerte. Für die bisherigen Untersuchungen wurde nur eine gestörte Kante betrachtet. Sind hingegen mehrere Kantengewichte ungültig, so steigt damit die Anzahl der gestörten FM, wenngleich dieselbe FM nur einmal gestört werden kann. Dieser Zusammenhang ist stark nicht-deterministisch, weil er von der relativen Lage des Baumes und den gestörten Kanten abhängt. Um dennoch einen konsistenten Graphen mit einer maximalen Anzahl von Kanten wie in Abb. 5.4(b) zu erhalten, werden im folgenden Kapitel unterschiedliche Ansätze vorgeschlagen.

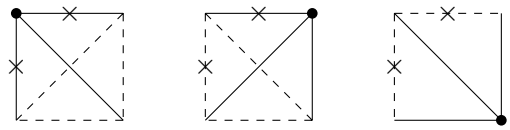
5.4 Lösungsansätze für die Synthese maximal möglicher, konsistenter Graphen

5.4.1 Iterieren des Wurzelknoten

Für die Berechnung des aufspannenden Baumes spielt die Wahl des Wurzelknotens eine wichtige Rolle. Zunächst wird durch geschicktes Iterieren des Wurzelknotens versucht, die ungültigen Kanten in den Kobaum zu legen und dadurch einen maximal möglichen Graphen zu synthetisieren. Wie in Kap. 5.3.2 diskutiert legen wir eine BFS-Suche zugrunde und nehmen zunächst an, dass der Graph vollständig zusammenhängend ist. Ist nur eine Kante ungültig und nicht im Kobaum, so müssen mindestens zwei weitere Wurzelknoten gewählt werden, damit die ungültige Kante sicher im Kobaum liegt. Dies veranschaulicht das Beispiel in Abb. 5.6(a). Hier ist der Wurzelknoten mit • und die ungültige Kante mit × gekennzeichnet. Es können maximal die zwei aufspannenden Bäume die ungültige Kante beinhalten, deren Wurzelknoten jeweils ein Endknoten der Kante ist. Deshalb reichen drei unterschiedliche aufspannende Bäume aus, um die gestörte Kante mindestens einmal in einen Kobaum zu legen.



(a) Eine ungültige Kante in einem vollständig zusammenhängenden Graphen

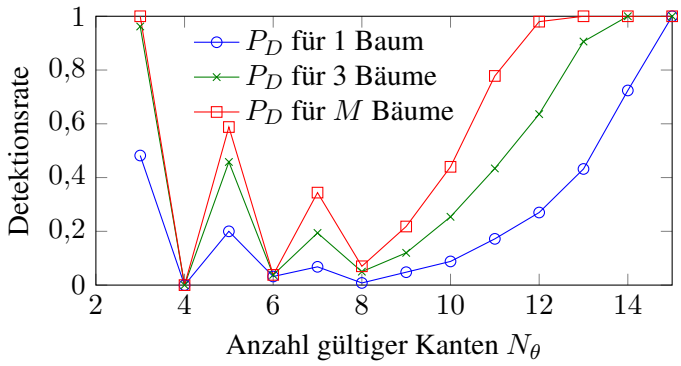


(b) Mehrere ungültige Kanten in einem vollständig zusammenhängenden Graphen

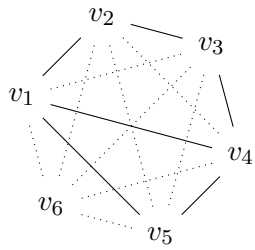
Abbildung 5.6: Minimale Anzahl unterschiedlicher Wurzelknoten, um eine ungültige Kante zu umgehen

Betrachtet man den Fall, dass mehrere Kanten fehlerhaft sind, wie in Abb. 5.6(b) gezeigt, so ist der Fall der Gleiche, wobei die fehlerhaften Kanten nacheinander in den Kobaum gelegt werden. Angenommen, die Kanten $e_1 = \{v_0, v_1\}$ und $e_2 = \{v_0, v_2\}$ sind fehlerhaft und es gilt, dass der erste Wurzelknoten v_0 ist. Dann folgt o.B.d.A. mit dem zweiten Wurzelknoten v_1, dass die fehlerhafte Kante e_2 bereits im Kobaum und mit einem dritten Wurzelknoten e_1 im Kobaum ist.

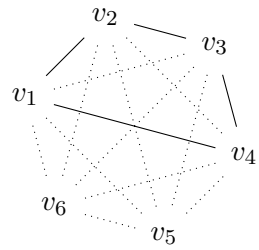
In Abb. 5.7(a) ist die Verbesserung durch mehrfaches Anwenden des SONG-Algorithmus



(a) Verbesserung der Detektionsrate bei mehreren ungültigen Kantengewichten durch wiederholte Berechnung eines aufspannenden Baumes



(b) Häufigste Topologie bei $N_\theta = 6$



(c) Häufigste Topologie bei $N_\theta = 4$

Abbildung 5.7: Synthese mit mehreren aufspannenden BFS Bäumen mit unterschiedlichen Wurzelknoten für einen Eingangsgraphen mit $M = 6$

mit unterschiedlichen Wurzelknoten für BFS-Bäume zu sehen. Dabei wurde die Detektionsrate über $S = 500$ unterschiedliche konsistente Teilgraphen gemittelt. Wie erwartet werden mehr konsistente Teilgraphen gefunden, wenn die Synthese von mehreren Knoten aus gestartet wird. Der Einbruch der Detektionsrate bei $N_\theta = 4, 6, 8$ Kantengewichten ist durch die Topologie der gültigen Kanten erklärbar. Alle generierten Graphen sind konsistente zweifach-zusammenhängende Teilgraphen, wie die durchgezogenen Kanten in Abb. 5.7(b) oder Abb. 5.7(c) zeigen. Die gepunkteten Kanten deuten die ungültigen Kantengewichte an. Bei dieser Topologie ist es nicht möglich, mit einem aufspannenden BFS-Baum die Masche v_1, v_2, v_3, v_4 nachzubilden, weil in einem vollständig zusammenhängenden Eingangsgraphen mit BFS immer Maschen der Länge drei gebildet werden.

Es ist zu erkennen, dass das Iterieren über drei Wurzelknoten bereits eine gute Verbesserung erzielt. Allerdings bringt eine Iteration über alle M Knoten nochmals einen deutli-

chen Gewinn für $N_\theta < 14$. Die führt erwartungsgemäß auch zu einem linearen Anstieg der Laufzeit wegen dem M-fachen Aufruf von SONG. Hinzu kommt der Schritt des Zusammenführens der unterschiedlichen Lösungsmengen. Die Synthese wird somit für das Iterationsverfahren über mehrere BFS-Bäume langsamer, aber genauer.

5.4.2 Aufspannender Baum mit A-priori-Wissen

Wie in Kap. 5.3.1 gezeigt ist ein Qualitätsmaß für die Kanten unumgänglich, um die Synthese partiell konsistenter Graphen weiter zu verbessern. Ist eine solche A-priori-Information verfügbar, kann diese für eine zielgerichtete Suche nach dem aufspannenden Baum genutzt werden. Damit wird begünstigt, dass möglichst viele ungünstige Kanten in den Kobaum verschoben werden. Das bekannteste Verfahren für einen qualitätsorientierten, aufspannenden Baum ist das Verfahren von Kruskal und Prim [Cormen u. a., 2009]. Der Algorithmus von Kruskal und Prim ist in Alg. 8 dargestellt.

Algorithmus 8: Aufspannender Baum nach Kruskal und Prim

Input : Knotenmenge V, Kantenmenge $E^q = (v_i, v_j, q)$
Output : Kantenmenge T, Knotenlabelmenge $\underline{L}$

```
L = [1, 2, ..., M];                          // jeder Knoten bekommt ein eigenes Label
Ē^q = qsort(E^q) ;                           // sortiere die Kanten absteigend nach q(v_i, v_j)
while Ē^q ≠ ∅ ∧ V ≠ ∅ do                     // es sind nicht alle Knoten abgearbeitet
    (v_i, v_j) = Ē^q(q_max);
    if L(v_i) ≠ L(v_j) then                  // falls die Labels der Endknoten unterschiedlich
        T = T ∪ (v_i, v_j);                  // füge die Kante an den aufspannenden Baum
        V = V \ {v_i, v_j};                  // entferne besuchte Endknoten
        for v ∈ [1, ..., M] do
            if L(v) == L(v_j) then
                L(v) = L(v_i);               // alle auf gemeinsames Label von v_i
            end
        end
    end
    Ē^q = Ē^q \ (v_i, v_j) ;
end
```

Zunächst bekommen alle Knoten eine unterschiedliche Baumnummer bzw. Label $\underline{L}$. Danach werden die Kanten $e = \{v_i, v_j\}$ entsprechend eines Qualitätsmaßes $q(v_i, v_j)$ sortiert und in absteigender Reihenfolge abgearbeitet. Dieses Qualitätsmaß sollte dabei die Verlässlichkeit der Kantengewichte berücksichtigen. Für die TDOA-basierte Lokalisierung kann dies der Betrag der GCC-PHAT an der Stelle der TDOA Schätzung sein, weil er ein Maß für die Gleichheit der Signale ist. Je höher die Ähnlichkeit, desto verlässlicher ist der geschätzte TDOA Wert. Haben die Endknoten v_i und v_j der aktuell untersuchten Kante unterschiedliche Label, so weist das darauf hin, dass sie aus bis-

her unterschiedlichen Bäumen stammen. Zu Beginn ist jeder Knoten sein eigener Baum und die Label sind unterschiedlich. Nun wird die Kante mit der höchsten Qualität ausgewählt und falls ihre Endknoten unterschiedliche Label aufweisen, wird sie zum aufspannenden Baum hinzugefügt. Die Endknoten werden als besucht von der Knotenmenge entfernt und alle Knoten mit dem gleichen Label wie v_i oder v_j auf ein einheitliches Label – $\underline{L}(v_i)$ in Alg. 8 – gesetzt. Dies entspricht einer Kombination der zwei Bäume von v_i und v_j. Falls die Endknoten v_i und v_j das gleiche Label haben, heißt es, dass sie bereits im gleichen Baum sind. Danach wird die Kante verworfen und die nächste Kante mit maximalem Qualitätsmaß abgearbeitet. Dies wird so lange wiederholt, bis alle Kanten einmal untersucht oder alle Knoten besucht wurden. Dadurch entsteht ein aufspannender Baum, der nur die Kanten mit maximaler Qualität beinhaltet.

5.4.3 Vergleich der Verfahren

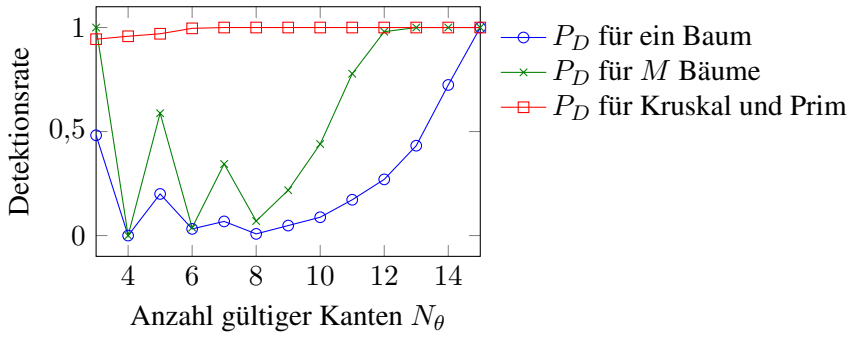
Generelle Unterschiede zwischen dem Iterationsverfahren des Wurzelknotens und dem aufspannenden Baum mit A-priori-Wissen sind in Tab. 5.2 festgehalten.

	Wurzelknoteniteration	A-priori-Wissen
Topologie	fest	variabel
Zusatzinformation	nein	ja
Parallelisierung	ja	nein
Aufwand	groß	gering

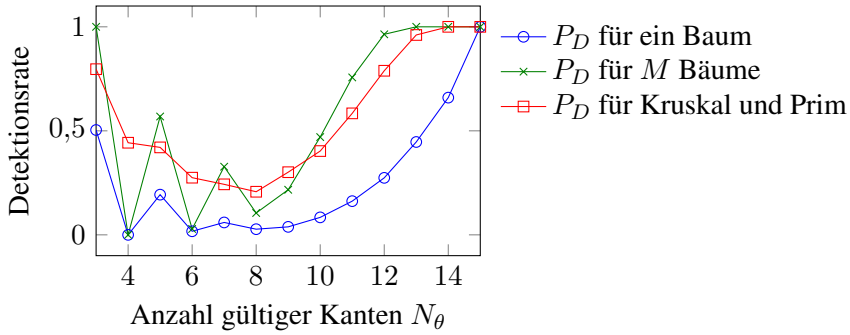
Tabelle 5.2: Unterschiede der Iteration des Wurzelknotens und des aufspannenden Baumes mit A-priori-Wissen nach Kruskal und Prim

In Abb. 5.8(a) werden die Detektionsraten der Ansätze für einen konsistenten Teilgraphen ($K = 1$) je Simulation mit N_θ gültigen Kanten verglichen. Die Detektionsraten sind wieder über $S = 500$ generierte Graphen gemittelt. Dabei wurde das Qualitätsmaß einer Kante (v_i, v_j, q) für den aufspannenden Baum von Kruskal und Prim nach folgender Regel berechnet: $q = 1$ bei einer gültigen Kante und $q = 0$ bei einer Ungültigen. Es existieren somit N_θ Kanten mit $q = 1$. Wie bereits aus Abb. 5.7(a) bekannt, sind die Einbrüche bei $N_\theta = 4, 6, 8$ für das Wurzelknoteniterationsverfahren zu sehen, welche auf die Topologie und die damit verbundenen notwendigen Längen der FM zurückzuführen sind. Das Verfahren von Kruskal und Prim kann einen nahezu perfekten aufspannenden Baum generieren und alle schlechten Kanten in den Kobaum verschieben. Dadurch erhält man eine besonders gute Detektionsrate.

In realen Anwendungen haben die unterschiedlichen, gesuchten Graphen allerdings unterschiedliche ungültige Kanten. Deshalb wurden für Abb. 5.8(b) fünf topologisch unterschiedliche Graphen generiert und in der gleichen Simulation vermischt. Das Qualitätsmaß je Kante q wurde hierbei als die Anzahl der gültigen Verwendungen berechnet,



(a) $K = 1$ konsistenter Graph je Simulation



(b) $K = 5$ konsistente Graphen je Simulation

Abbildung 5.8: Detektionsraten der unterschiedlichen konsistenten Teilgraphen in einem vollständig zusammenhängenden Graphen mittels der Syntheseverfahren mit unterschiedlichen aufspannenden Bäume.

d.h. wenn die Kante in drei der fünf konsistenten Teilgraphen verwendet wurde, so ist $q = 3$. Es gilt $\sum_{e \in E^q} q(e) = 5N_\theta$. Man sieht deutlich, dass das Verfahren nach Kruskal und Prim einen Einbruch aufweist, da es nur für eine Lösung einen optimalen aufspannenden Baum generieren kann. Dieser ist für die anderen gesuchten Teilgraphen selten optimal. Hier ist wiederum der M-fache BFS-Baum von Vorteil, der für die unterschiedlichen gesuchten Graphen jeweils einen möglichst guten Baum generiert und damit ab $N_\theta \geq 10$ bessere Detektionsraten aufweisen kann.

Kapitel 6

Anwendung in der Sprecherlokalisierung

Wie eingangs erwähnt, stellt die TDOA-basierte akustische Sprecherlokalisierung mit mehreren Quellen und in reflektionsbehafteten Umgebungen eine Problemstellung dar, die mittels der Synthese konsistenter Graphen gelöst werden kann. Dafür werden in diesem Kapitel zuerst die Probleme bei der zeitdiskreten Verarbeitung der akustischen Signale genauer analysiert und anschließend wird auf die Anpassung des SONG-Algorithmus auf diesen Anwendungsfall eingegangen. Hierbei werden wir zuerst SONG in Simulationen analysieren und abschließend in realen Messungen anwenden.

6.1 Störungen in der TDOA-Schätzung

6.1.1 Grenzen der Lokalisierbarkeit und Signaleigenschaften

Bei akustischen Sprachsignalen handelt es sich um ein breitbandiges Signal zwischen 100Hz–8kHz [Vary u. a., 1998] mit gemeinsamen Frequenzbereichen zwischen unterschiedlichen Quellsignalen. Außerdem führt die inhomogene Temperatur- und Feuchtigkeitsverteilung im Ausbreitungsmedium zu frequenzabhängigen Laufzeiten [Scheuing, 2007]. Aufgrund der räumlichen Lage der Mikrofone ist die maximal messbare und darstellbare Frequenz des aufgezeichneten Signals nach dem räumlichen Nyquist-Kriterium begrenzt: zwischen der Wellenlänge λ und dem Abstand d der Mikrofone muss die Bedingung $d \leq \frac{\lambda}{2}$ erfüllt sein, um kein räumliches Aliasing zu erhalten. Beispielhaft liegt für 1kHz der Maximalabstand bei 17cm. Für breitbandige Signale können quasi beliebig verteilte Mikrofone verwendet werden [Brandstein u. Silverman, 1997]. Meist liegen die energiereichen Frequenzen für Sprachsignale im Bereich von 100Hz bis 1kHz, wie Abb. 6.1 für ein weibliches und ein männliches Sprachsignal zeigt.

6.1.2 Genauigkeit der TDOA-Schätzung

Neben den Aliasingfehlern, die durch geeignete Mikrofonpositionierung beseitigt werden, ergeben sich Probleme zum einen bei der zeitdiskreten Verarbeitung und zum anderen durch Hintergrundrauschen an den Mikrofonen.

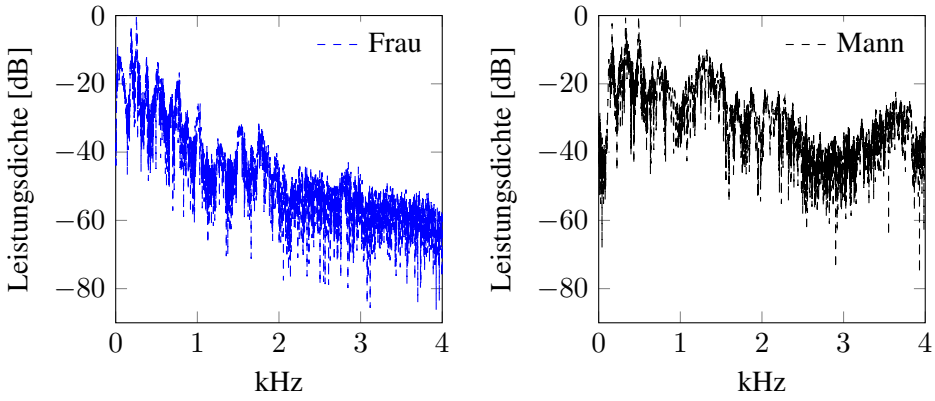


Abbildung 6.1: Spektrale Leistungsdichteverteilung für zwei reale Sprachsignale

Zeitdiskrete Verarbeitung der Kreuzkorrelation

Abtastung Wegen der Verarbeitung in einem PC werden die Mikrofonsignale über einen Analog-Digital-Wandler mit der Abtastfrequenz f_A abgetastet. Dabei muss nach Nyquist die Abtastfrequenz für Sprachsignale $f_A \geq 16\text{kHz}$ sein. Nach äquidistanter Abtastung liegen die Mikrofonsignale

$$x_m[n] = x_m(n \cdot T_A) \tag{6.1}$$

mit dem Abtastintervall $T_A = \frac{1}{f_A}$ vor [Oppenheim u. Schafer, 1999]. Die zeitdiskrete Kreuzkorrelation ist

$$r_{ij}[n] = \lim_{B\to\infty} \frac{1}{2B} \sum_{\mu=-B+1}^{B} x_i[n+\mu]x_j[\mu] = r_{ij}(nT_A) \tag{6.2}$$

mit der Blocklänge B [Scheuing, 2007]. In einer realen Anwendung können aber nur kausale Systeme betrachtet werden, weshalb allgemein eine blockweise Verarbeitung mit endlicher Länge angewendet wird. Dies führt zur zeitdiskreten Kurzzeitkreuzkorrelation [Vary u. a., 1998]

$$\bar{r}_{ij}[b,n] = \frac{1}{B} \sum_{\mu=b-B+1}^{b} x_i[n+\mu]x_j[\mu], \tag{6.3}$$

welche einer Multiplikation mit einem Rechteckfenster von $n = b - B + 1$ bis $n = b$ entspricht. Zur Vereinfachung wird nachfolgend der Faktor $\frac{1}{B}$ weggelassen, da der qualitative Verlauf der Kreuzkorrelation entscheidend ist. Außerdem kann vereinfachend der Laufindex b weggelassen werden, indem der Block bei n beginnt und bis $n + B - 1$

geht. Zur effizienteren Berechnung der Kurzzeitkreuzkorrelation wird diese im Frequenzbereich durchgeführt. Dabei werden $x_i[n]$ und $x_j[n]$ mittels der diskreten Fourier-Transformation (DFT) in den diskreten Frequenzbereich transformiert. Um eine zyklische Korrelation zu vermeiden, muss hierbei Zero-Padding angewendet werden, wobei die Zeitsignale mit Nullen erweitert werden. Anschließend erhält man im diskreten Frequenzbereich die Signale $X_i[k]$ und $X_j[k]$ und nach einer inversen DFT (IDFT) die zeitdiskrete GCC

$$r_{ij}[n] = \mathrm{IDFT}\left(X_i[k]X_j^*[k]\right) \tag{6.4}$$

mit dem Filter $\Psi[k] = 1$. Eine perfekte Rekonstruktion der kontinuierlichen Kreuzkorrelation erfolgt durch

$$r_{ij}(\tau) = \sum_{n=-\infty}^{\infty} r_{ij}[n]g(t - nT_A), \quad g(t) = \mathrm{sinc}\left(\frac{\pi t}{T_A}\right) = \begin{cases} \frac{\sin(\pi t/T_A)}{(\pi t/T_A)}, & t \neq 0 \\ 1, & t = 0 \end{cases}. \tag{6.5}$$

Da nur kausale Systeme verwirklicht werden, ergibt sich ein Quantisierungsfehler zwischen der korrekten Maximumschätzung nach (2.56) und dem abgetasteten Wert bei

$$\hat{n}_{ij} = \arg\max_n \; r_{ij}[n]. \tag{6.6}$$

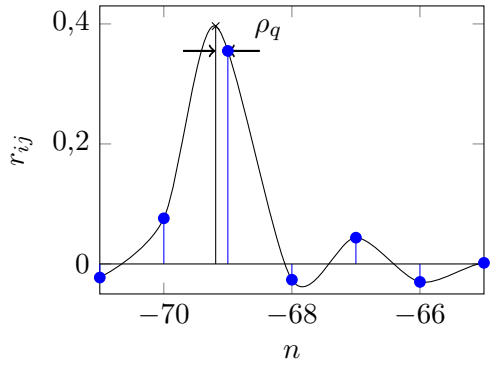


Abbildung 6.2: TDOA Quantisierungsfehler durch diskrete zeitliche Abtastung der GCC-PHAT

Dies veranschaulicht Abb. 6.2, die den Versatz des richtigen TDOA-Maximums ($\times$) von dem abgetasteten Maximum ($\bullet$) zeigt. Diese Differenz beschreibt den Quantisierungsfehler ρ_q, der bei der zeitdiskreten Verarbeitung entsteht.

Interpolation der TDOA-Schätzung Weil eine perfekte Rekonstruktion nicht möglich ist, wird mit einer quadratischen Interpolation die Schätzung der Maxima verbes-

sert. Dabei wird der geschätzte Zeitindex $\hat{n}_{ij}$ der TDOA mit dem jeweils vorhergehenden und nachfolgenden Abtastwert verglichen und mit

$$\tilde{n}_{ij} = \hat{n}_{ij} + \frac{r_{ij}(\hat{n}_{ij} - 1) - r_{ij}(\hat{n}_{ij} + 1)}{2r_{ij}(\hat{n}_{ij} - 1) - 4r_{ij}(\hat{n}_{ij}) + 2r_{ij}(\hat{n}_{ij} + 1)} \tag{6.7}$$

verbessert [Drews, 1995]. Der Einfluss dieser Methode wurde durch eine Monte-Carlo-Simulation verifiziert, deren Ergebnis in Tab. 6.1 gezeigt ist. Dabei wurde ein Rauschsignal an der Quelle generiert und mit der über die Spiegelmethode (Image-Source-Model, ISM) von [Lehmann u. Johansson, 2008] simulierten Impulsantwort zu jedem Mikrofon gefaltet. Die Nachhallzeit war mit $T_{60} = 200$s kurz und der Raum hatte die Abmessungen $5 \times 5 \times 3\text{m}^3$. Dies wurde für 50 zufällig positionierte Mikrofonarrays mit $M = 5$ Mikrofonen durchgeführt und anschließend die TDOA aus jeweils 20 Signalblöcken für alle Mikrofonpaare ermittelt. Die Abtastfrequenz betrug $f_A = 16$kHz und die Blocklänge $B = 2048$ Abtastwerte. Der Fehler, der sich ohne die Interpolation ergibt, ist $\hat{\rho}_{ij} = t_{ij} - \hat{n}_{ij}T_A$ und mit Interpolation $\tilde{\rho}_{ij} = t_{ij} - \tilde{n}_{ij}T_A$. Der mittlere Fehler $\bar{\mu}$ berechnet sich nach

$$\bar{\mu} = \frac{1}{S}\sum_{S}\frac{1}{N}\sum_{ij}\hat{\rho}_{ij} \tag{6.8}$$

bzw. $\frac{1}{S}\sum_S \frac{1}{N}\sum_{ij}\tilde{\rho}_{ij}$ über alle Simulationen $S = 50 \cdot 20$. Die mittlere Standardabweichung $\bar{\sigma}$ ist ebenso über alle Standardabweichungen $\hat{\sigma}_{ij}$ gemittelt. Es ist zu sehen, dass die Interpolation in Abwesenheit von Rauschen den mittleren Fehler stark reduziert. Die mittlere Standardabweichung bleibt hingegen nahezu gleich. Wird zu den Mikrofonsignalen ein weißes Gaußsches Rauschen mit einem SNR$= \sigma^2_{x_i}/\sigma^2_{\rho_i} = 0$dB addiert, so erhält man einen nahezu gleichen mittleren Fehler für beide Verfahren, der logischerweise größer als im rauschfreien Fall ist. Dabei ist $\sigma^2_{x_i}$ die Leistung der Mikrofonsignale $x_i[n]$ und $\sigma^2_{\rho_i}$ die Rauschleistung. Die mittleren Standardabweichungen bleiben auch in diesem Fall nahezu gleich.

ohne Rauschen	ohne Interpolation	mit Interpolation
mittlerer Fehler $\bar{\mu}$	$0,0049\,T_A$	$0,0001\,T_A$
mittlere Standardabweichung $\bar{\sigma}$	$2,348\,T_A$	$2,331\,T_A$
mit Rauschen		
mittlerer Fehler $\bar{\mu}$	$0,2062\,T_A$	$0,2026\,T_A$
mittlere Standardabweichung $\bar{\sigma}$	$8,893\,T_A$	$8,891\,T_A$

Tabelle 6.1: Einfluss der quadratischen Interpolation auf den mittleren TDOA-Schätzfehlers mit und ohne additiven weißen Gaußschen Rauschen

Einfluss der Abtastfequenz Eine Interpolation kann ausgelassen werden, wenn eine höhere Abtastfrequenz gewählt wird. Damit ist neben einem höheren Rechenaufwand

auch eine Verringerung des zeitlichen Quantisierungsfehlers ρ_q der TDOA-Schätzung verbunden. Der Fehler ist bei $f_A = 16\text{kHz}$ maximal $\frac{T_A}{2} = \pm 0,03\text{ms}$ und bei $f_A = 48\text{kHz}$ nur noch $\pm 0,01\text{ms}$.

Analyse des TDOA-Schätzfehlers

Zu dem Quantisierungsrauschen ρ_q addiert sich in realen Anwendungen noch das weiße Gaußsche Hintergrundrauschen ρ_i an den Mikrofonen dazu. Es wird in [Quazi, 1981] gezeigt, dass für geringe und identische SNR-Werte der resultierende TDOA-Fehler ρ_{ij} einer Normalverteilung $\mathcal{N}(\mu_{ij}, \sigma_{ij}^2)$ mit

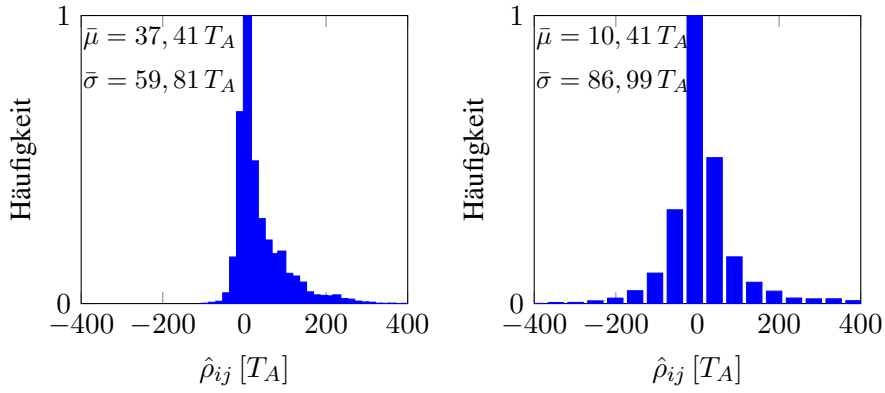
$$\sigma_{ij}^2 \sim \frac{1}{\text{SNR}^2} \frac{1}{B} \frac{1}{f_0^2} \frac{1}{W} \tag{6.9}$$

folgt. Dabei ist W die Bandbreite und f_0 die Mitte des betrachteten Frequenzbandes. Nimmt man weiterhin wie in [So u. a., 2008] unkorreliertes Rauschen mit identischem SNR an jedem Mikrofonpaar an, dann ist

$$\sigma_{ij}^2 = \frac{3(1 + 2\text{SNR})}{\pi^2 B \text{SNR}^2} \tag{6.10}$$

gegeben. Allerdings ist die Signalenergie der Quellen zu den Mikrofonen stark abhängig vom Abstand d mit d^{-2} [Goelzer u. a., 1995], was besonders im Nahfeld zu unterschiedlichen Varianzen entlang der einzelnen Mikrofonpaare führt. Die lange Impulsantwort einer hohen Nachhallzeit von $T_{60} \geq 400\text{ms}$ lässt zudem nur Abschätzungen der statistischen Verteilung des Rauschens zu. Deshalb werden wir uns im Weiteren auf Simulationen beziehen.

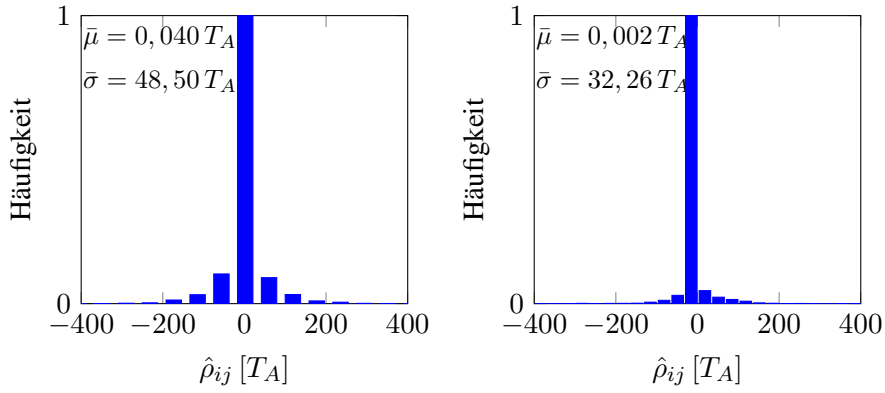
Abbildung 6.3 zeigt die Histogramme der Fehler $\hat{\rho}_{ij}$ ohne die quadratische Interpolation für eine Simulation mit der ISM nach [Lehmann u. Johansson, 2008]. Das Quellsignal ist ein Rauschsignal, die Nachhallzeit ist $T_{60} = 400\text{ms}$ und das $\text{SNR} = 0\text{dB}$. Die resultierenden Fehler wurden wieder über 50 zufällige Mikrofonarrays mit $N = 10$ Mikrofonpaaren und 20 Signalblöcke der Länge $B = 128\text{ms}$ gemittelt. Zunächst erkennt man, dass die mittleren Fehler in der TDOA-Schätzung für eine realitätsnahe Simulation nicht mittelwertfrei sind und in Abb. 6.3 relativ hoch im Vergleich zu Tab. 6.1. Das lässt sich zum Teil durch die kurze Blocklänge im Vergleich zu der langen Nachhallzeit erklären. Des weiteren ist zu erkennen, dass der mittlere Fehler $\bar{\mu}$ bei einer dreifach größeren Abtastfrequenz auch ungefähr dreimal so klein ist. Die Varianz hat sich dagegen leicht vergrößert, allerdings nicht in einem multiplikativen Maß, sondern geringer. Da in Abb. 6.3 nur der Fehler bezogen auf das Abtastintervall gezeigt ist, muss man ihn noch mit diesem multiplizieren um auf den realen Fehler zu kommen. Aus den Simulationen ergibt sich, dass in verrauschten Umgebungen die Erhöhung der Abtastfrequenz gegenüber der quadratische Interpolation eine Verbesserung des mittleren Fehlers bringt, verbunden mit einem höheren Rechenaufwand.



(a) $f_A = 16$kHz (b) $f_A = 48$kHz

Abbildung 6.3: Histogramme des TDOA-Fehlers $\hat{\rho}_{ij}$ bei unterschiedlichen Abtastfrequenzen, $T_{60} = 400$ms, SNR $= 0$dB und Blocklänge $B = 128$ms

Einfluss der Blocklänge Um den Einfluss der Blocklänge zu untersuchen, wurde die gleiche Simulationsumgebung wie für Abb. 6.3 verwendet mit einem Rauschsignal an der Quelle, $T_{60} = 400$ms und SNR$= 0$dB. Allerdings wurde zunächst eine Blocklänge von $B = 4\,800$ und anschließend von $B = 24\,000$ Abtastwerten bei $f_A = 48$kHz gewählt. Abbildung 6.4 zeigt, wie die Schätzung der TDOA aus der Kurzzeitkreuzkorrelation mit zunehmender Blocklänge exakter wird.



(a) $B = 4\,800 \hat{=} 100$ms (b) $B = 24\,000 \hat{=} 500$ms

Abbildung 6.4: Histogramme des TDOA-Fehlers $\hat{\rho}_{ij}$ für unterschiedliche Blocklängen bei $f_A = 48$kHz, $T_{60} = 400$ms und SNR $= 0$dB

Es muss darauf geachtet werden, dass die Blocklänge groß genug gewählt wird, um eine zuverlässige Schätzung zu gewährleisten [Hahn u. Tretter, 1973], aber klein genug,

um bewegende Quellen zu erkennen und eine Echtzeitverarbeitung zu ermöglichen. In [Gerhard, 2006] werden dazu sehr kurze Blocklängen von 30ms gewählt. In einem solchen Fall ist bei starkem Hall die lange Raumimpulsantwort über mehrere Signalblöcke verteilt. Da in dieser Arbeit reflexionsbehaftete Umgebungen betrachtet werden, wird eine Blocklänge größer 200ms gewählt.

Schlussendlich muss bei der Blocklänge noch sichergestellt werden, dass der Mikrofonabstand und der damit verbundene maximale TDOA-Wert in den zuverlässigen Bereich der Blocklänge passt: $|\tau_{\max}| \leq \frac{2}{3} B T_A$ [Scheuing, 2007].

Fensterfunktion Durch die blockweise Verarbeitung der Signale entstehen Artefakte durch die einhergehende Faltung im Frequenzbereich

$$x_m[n] \cdot w[n] \overset{\text{DFT}}{\underset{\text{IDFT}}{\rightleftarrows}} X[k] * W[k] \tag{6.11}$$

mit der Fensterfunktion

$$w[n] = \begin{cases} 1, & 0 \leq n < L \\ 0, & \text{sonst} \end{cases}. \tag{6.12}$$

Diese einfache Rechteckfunktion lässt sich als Fourier-Transformierte

$$W\left(\mathrm{e}^{\mathrm{j}\omega}\right) = \mathrm{e}^{-\mathrm{j}\omega\frac{L-1}{2}} \cdot \begin{cases} \frac{\sin(L\omega/2)}{\sin(\omega/2)} & \omega \neq 2\pi k,\ k \in \mathbb{Z} \\ L & \omega = 2\pi k,\ k \in \mathbb{Z} \end{cases} \tag{6.13}$$

darstellen. Die dabei enstehende Welligkeit kann mithilfe von unterschiedlichen Fensterfunktionen wie Dreieck, Hamming oder Hann gemindert werden, was aber zu einer Verschlechterung der zeitlichen Auflösung führt [Oppenheim u. Schafer, 1999]. Dafür werden die einzelnen Signalblöcke um jeweils einen halben Signalblock davor und danach erweitert und anschließend im Zeitbereich mit dem jeweiligen Fenster der Länge $2B$ elementweise multipliziert. Für die Simulationen wurde ein Rauschen als Quellsignal mit $B = 250$ms bei $f_A = 48$kHz gewählt und wiederum 50 unterschiedliche Mikrofonarrays mit 20 Signalblöcken verarbeitet. Die Nachhallzeit in der Simulation betrug $T_{60} = 400$ms und das SNR$= 0$dB. Der mittlere Fehler wurde nach (6.8) berechnet.

Fenster	Rechteck	Hamming	Hann
mittlerer Fehler $\bar{\mu}$	$0,302\,T_A$	$0,168\,T_A$	$0,075\,T_A$
Standardabweichung $\bar{\sigma}$	$19,98\,T_A$	$24,68\,T_A$	$25,77\,T_A$

Tabelle 6.2: Mittlerer Fehler der TDOA-Schätzung bei unterschiedlicher Fensterfunktion

Tabelle 6.2 zeigt eine Verbesserung des mittleren Fehlers vom Rechteck-Fenster über Hamming hin zum Hann-Fenster. Allerdings ist auch eine Vergrößerung der Standardabweichung zu beobachten. Die Ergebnisse liegen somit alle in einem vergleichbaren Bereich.

Phasentransformationsfilter Zuletzt betrachten wir die Auswirkungen des PHAT-Filters, der nach [Knapp u. Carter, 1976] für die TDOA-Schätzung gut geeignet ist. Hierfür wird (6.4) wie folgt umformuliert

$$r_{ij}[n] = \mathrm{IDFT}\left(\frac{X_i[k]X_j[k]}{|X_i[k]X_j^*[k]|}\right). \tag{6.14}$$

Der PHAT-Filter sorgt dafür, dass alle Phaseninformationen gleich gewichtet werden, indem ihre zugehörigen Amplituden der Kreuzkorrelation normiert werden.

In Tab. 6.3 sind die Ergebnisse aus den Simulationen mit $B = 500\text{ms}$, $T_{60} = 400\text{ms}$ und SNR$= 0$dB gezeigt. Die Ergebnisse wurden wieder über 50 Mikrofonanordnungen und jeweils 20 Signalblöcke gemittelt. Zunächst ist der mittlere Fehler und die Standardabweichung für ein Rauschsignal an der Quelle gegeben. Es zeigt sich, dass in diesem Fall der PHAT-Filter eine leichte Verschlechterung der Ergebnisse bewirkt. Ersetzt man aber das Quellsignal durch ein männliches Sprachsignal, so erhält man eine Verbesserung durch den PHAT-Filter, wie die Ergebnisse in Tab. 6.3 zeigen.

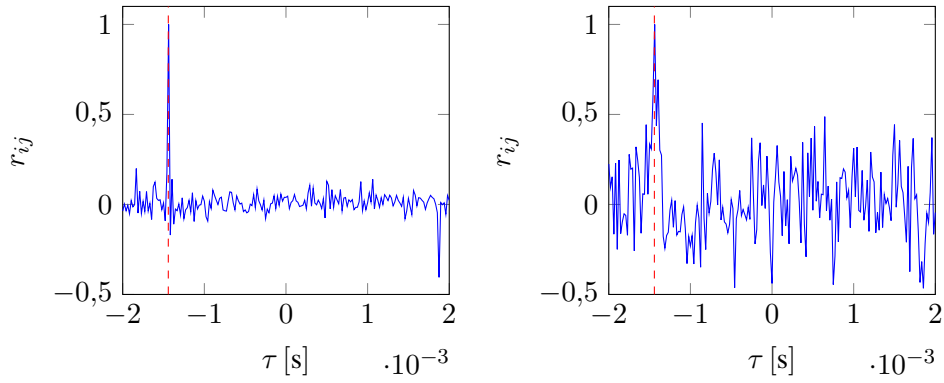
	ohne PHAT	mit PHAT
Rauschsignal	$\Psi[k] = 1$	$\Psi[k] = \Psi_{\mathrm{PHAT}}$
mittlerer Fehler $\bar{\mu}$	$0,177\,T_A$	$-0,371\,T_A$
Standardabweichung $\bar{\sigma}$	$25,30\,T_A$	$29,58\,T_A$
Sprachsignal		
mittlerer Fehler $\bar{\mu}$	$-3,142\,T_A$	$-0,914\,T_A$
Standardabweichung $\bar{\sigma}$	$159,16\,T_A$	$77,61\,T_A$

Tabelle 6.3: Verbesserung des mittleren TDOA-Schätzfehlers durch den Phasentransformationsfilter

6.1.3 Signalspezifische Mehrdeutigkeiten

Neben den in Kap. 2.3.1 erwähnten Problemen wie Mehrwegeausbreitung und Mehrquellenszenarien erhält man noch weitere Störeinflüsse, die zu nicht eindeutigen TDOA-Zuweisungen führen.

Einfluss des Quellsignals Eine weitere Besonderheit von Musik und Sprachsignalen sind ihre Periodizitäten aufgrund von Formanten [Vary u. a., 1998]. Diese Periodizitäten führen bei der Berechnung der GCC-PHAT zu wiederholten Maxima [Scheuing, 2007]. So führt ein weißes Rauschsignal mit einem konstanten Leistungsdichtespektrum zu einer GCC-PHAT wie in Abb. 6.5(a) und ein Sprachsignal zu einer GCC-PHAT wie in Abb. 6.5(b).



(a) GCC-PHAT eines Rauschsignals (b) GCC-PHAT eines Sprachsignals

Abbildung 6.5: Simulierte GCC-PHAT für unterschiedliche Quellsignale mit $T_{60} = 400$ms und SNR$= 20$dB

Es ist gut zu erkennen, dass die Rauschsignale durch ihren eindeutigen zeitlichen Verlauf eine besonders ausgeprägte Charakteristik in der Kreuzkorrelation aufweisen. Darum ergibt sich ein besonders schmaller Peak an der korrekten TDOA-Position. Für ein echtes Sprachsignal mit den Periodizitäten und einem nicht konstanten Leistungsdichtespektrum erhält man mehrere, nicht eindeutige Maxima. Dies erschwert die Schätzung des korrekten TDOA-Wertes besonders für Mehrquellen und Mehrwegeausbreitung.

In den bisherigen Simulationen wurden nur Rauschsignale betrachtet, um das Problem der Wahl eines falschen TDOA-Maxima zu minimieren. Außerdem tritt bei einem Rauschsignal das Problem von Sprachpausen nicht auf. Um Falschdetektionen in einer Sprachpause zu reduzieren, ist es in der Lokalisierung wichtig, diese zu erkennen und die Lokalisierung zu unterbrechen. Ein gängiges Verfahren zur Sprachpausenerkennung ist die Schätzung der Signalenergie aus den aufgezeichneten Mikrofonsignalen [Huang u. a., 2004]. Diese erfolgt nach

$$P_{\mathrm{VAD}} = \frac{1}{MB} \sum_{i=1}^{M} \|\underline{x}_i\|_p \tag{6.15}$$

mit $\underline{x}_i = [x_i[0], \ldots, x_i[B-1]]^T$ über alle M Mikrofonsignale. Die l_2-Norm ($p = 2$) liefert die Gesamtenergie aus allen Mikrofonsignalen und kann über eine initiale

Rauschsignalmessung das SNR berechnen. Für eine praktische Implementierung kann auch die l_1-Norm verwendet werden, welche einfacher zu berechnen ist und einzelne Frequenzbereiche weniger stark gewichtet [Kaschub, 2007].

Reduktion des Signalpegels Neben der starken Dämpfung des Schalldrucks zwischen Quelle und Mikrofon mit dem Abstand d^{-2}, ist auch jede weitere akustische Quelle eine Störung der anderen. Dies ist in Abb. 6.6 gezeigt. Hier ist die GCC für zwei unterschiedliche Rauschsignale dargestellt, die einzeln eine gute Korrelation zeigen. Wenn beide Quellen simultan aktiv sind, wie im unteren Diagramm gezeigt, ist die Signalamplitude der GCC zum einen insgesamt gedämpft und zum anderen ist die Korrelation der Quelle 2 schlechter ausgeprägt als im Einzelfall.

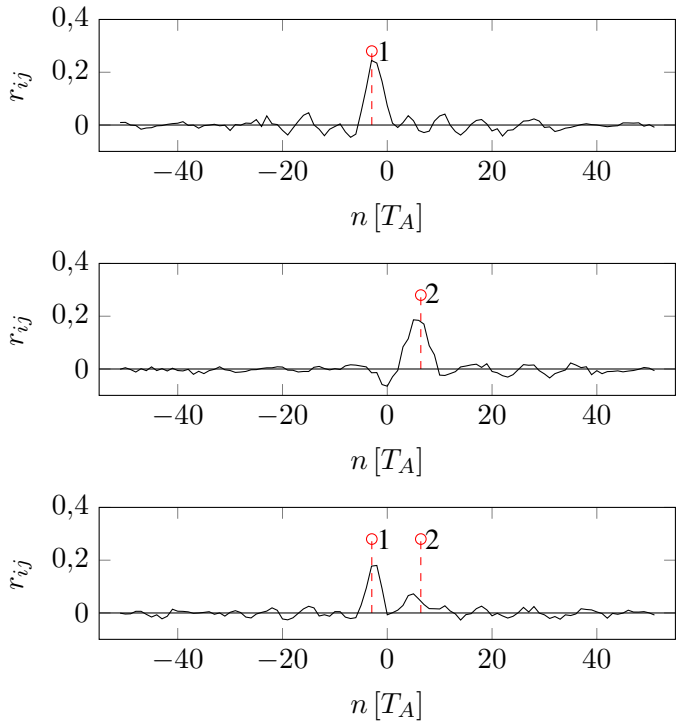


Abbildung 6.6: GCC für jeweils zwei Rauschquellen und bei gleichzeitiger Aktivität beider Quellen

6.2 SONG in realitätsnahen Simulationen

In diesem Kapitel wird die Synthese partiell konsistenter Graphen für den Fall mit verrauschten Kantengewichten angewendet. Dabei liegt immer ein BFS-Baum und ein

Wurzelknoten mit maximalem Knotengrad zugrunde, was, wie in Kap. 4 und Kap. 5 gezeigt, besonders effizient und robust ist.

6.2.1 Näherungsweise Konsistenz

Bei der Verwendung von SONG in einer realen Anwendung kommt es immer zu leichten Abweichungen der Kantengewichte von den exakten Werten. Dies beruht meist auf Messungenauigkeiten, Hintergrundrauschen der Sensoren oder den Quantisierungsfehlern bei der Abtastung der Signale, wie in Kap. 6.1.2 diskutiert. Um eine korrekte Funktionsweise von SONG zu gewährleisten, muss für diese Fälle die *näherungsweise Konsistenz* betrachtet werden. Diese berücksichtigt kleine Abweichungen von der *exakten Konsistenz*, die nach (3.6) genau Null ergeben muss. Es wird dabei angenommen, dass die Messwerte w_n durch einen normalverteilten Rauschprozess $\rho_n \sim \mathcal{N}(\mu_n, \sigma_n^2)$ mit dem Mittelwert $\mu_n = 0$ und Varianz σ_n^2 nach $w_n + \rho_n$ additiv überlagert werden.

Angenommen, die einzelnen Rauschprozesse entlang einer Masche sind statistisch unabhängig voneinander. Dann folgt die zyklische Summe einer Überlagerung von Gaußverteilungen nach $\rho_l = \sum_n \rho_n,\ (1 \leq n \leq N : \underline{l}_i(n) \neq 0)$. Da die einzelnen Rauschprozesse statistisch unabhängig sind und unter der weiteren Annahme, dass sie identische Leistungen haben, besitzt die Summe eine Varianz von $\sigma_l^2 = \sum_n \sigma_n^2 = N_i \sigma_n^2$ bzw. eine Standardabweichung von $\sigma_l = \sqrt{N_i}\sigma_n$, wobei $N_i = \|\underline{l}_i\|_0$ die l_0-Norm, d.h. die Anzahl der nicht-Null-Elemente darstellt. Daraus ergibt sich eine Schranke für die Konsistenzbedingung einer FM $\underline{l}_{f,i}$

$$|\underline{l}_{f,i}^T \underline{w}| \leq \epsilon \cdot \sqrt{N_i}. \tag{6.16}$$

Mit dem Parameter ϵ kann nun abhängig von der vorliegenden Rauschleistung σ_n^2 der Schwellwert angepasst werden, mit dem eine Masche konsistent ist. Ist ϵ klein, so werden nur wenige Kantengewichtskombinationen als näherungsweise konsistent akzeptiert. Diese können aber als sehr verlässlich angesehen werden. Wird ϵ zu groß gewählt, kann es sein, dass die vermeintlich konsistente Masche nicht auf den richtigen Kantengewichten mit geringen Ungenauigkeiten beruht, sondern durch eine ungewünschte Kombination zustande kommt.

In den folgenden Simulationen wurde zunächst nur ein konsistenter Graph mit $M = 6$ Knoten und $N = 15$ konsistenten Kantengewichten bestimmt. Die Kantengewichte wurden anschließend mit einem additiven, weißen Gaußschen-Rauschen mit der Varianz σ_n^2 überlagert. In Abb. 6.7 sind die Detektionsraten für unterschiedliche Schwellwerte ϵ gezeigt, gemittelt über $S = 1\,000$ simulierte Graphen. Die Detektionsrate beschreibt dabei wie oft der generierte konsistente Graph vollständig synthetisiert wurde. Sie zeigt wie erwartet eine besonders gute Detektion im rauschfreien Fall ($\sigma_n = 0$) und einen starken Abfall für ansteigendes Rauschen.

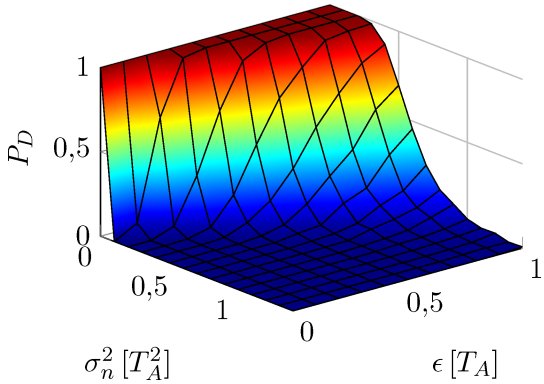


Abbildung 6.7: Detektionsraten für einen vollständig konsistenten Graphen mit unterschiedlichen Rauschvarianzen σ_n^2 und Schwellwerten ϵ

6.2.2 Falsche Konsistenz

Konsistenz aufgrund der Orthogonalität Die Orthogonalität zwischen den Spaltenräumen der Inzidenz- und Maschenmatrix führt zu der Eigenschaft, dass alle Spaltenvektoren $\underline{z}_i \in \mathbf{Z}$ aus (3.1) eine Lösung für die Nullsummenbedingung aus (3.6) darstellen, siehe Ende Kap. 3.2.2. Ist ein Graph mit den konsistenten Kantengewichten $\underline{w}_k$ gegeben, so ist

$$\underline{\tilde{w}}_k = \underline{w}_k + \sum_{i=1}^{M} \alpha_i \underline{z}_i \tag{6.17}$$

mit $\alpha_i \in \mathbb{R}$ ebenso eine gültige Lösung zu (3.6), da

$$\mathbf{L}_{\mathrm{FM}}^T \underline{\tilde{w}}_k = \mathbf{L}_{\mathrm{FM}}^T (\underline{w}_k + \sum_i \alpha_i \underline{z}_i) = \mathbf{L}_{\mathrm{FM}}^T \underline{w}_k + \sum_i \alpha_i \mathbf{L}_{\mathrm{FM}}^T \underline{z}_i = 0. \tag{6.18}$$

Dabei stellt $\alpha_i \underline{z}_i$ eine additive Änderung aller Kantengewichte dar, die mit dem i-ten Knoten verbunden sind.

Führt man das auf die Nullsummenbedingung (2.65) für die TDOA-basierte Lokalisierung zurück, so entspricht dies einer Änderung aller TDOA zu einem Mikrofon i. Da in (6.17) alle Kantengewichte (TDOA) zum Knoten (Mikrofon) i gleich verändert werden, gilt $\tilde{w}_{ij} = w_{ij} \pm \alpha_i$ für alle Kantengewichte zu i. Das bedeutet für die jeweiligen TDOA, dass alle $\tau_{ij,q,\mu 0} = t_{iq,\mu} - t_{jq,0}$ mit $t_{iq,\mu} = t_{iq,0} \pm \alpha_i$ geändert wurden [Kreißig u. Yang, 2010; Yang u. Kreißig, 2011]. Mit (6.17) können wir somit das Phänomen der Mehrwegeausbreitung für konsistente Graphen beschreiben. Allerdings bedeutet das im Umkehrschluss, dass die Konsistenzbedingung auch für die Mehrwegeausbreitung gilt und für die Direktpfad-TDOA nur eine notwendige aber nicht hinreichende Bedingung ist.

Es gilt zudem, dass die Überlagerung von konsistenten Kantengewichtskombinationen $\underline{\tilde{w}} = \sum_k \beta_k \underline{w}_k$ ebenso konsistent ist, denn

$$\mathbf{L}_{\mathrm{FM}}^T \underline{\tilde{w}} = \mathbf{L}_{\mathrm{FM}} \left(\sum_k \beta_k \underline{w}_k \right) = \sum_k \beta_k \underbrace{\mathbf{L}_{\mathrm{FM}}^T \underline{w}_k}_{=0} = 0. \tag{6.19}$$

Konsistenz aufgrund zusätzlicher Werte Im Gegensatz zu den obigen zusätzlichen konsistenten Lösungen, die aufgrund der Orthogonalitätbeziehung zwischen $\mathbf{L}_{\mathrm{FM}}$ und $\mathbf{Z}$ zustande kommen, ergeben sich auch zufällig neue konsistente Lösungen. Ein Beispiel hierfür ist in Abb. 6.8 gezeigt.

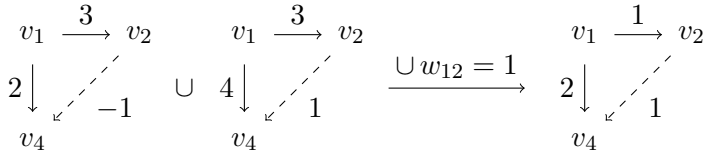


Abbildung 6.8: Ungewünschte zusätzliche Lösung

Durch eine weitere, fehlerhafte Messung $w_{12} = 1$ entsteht die neue, konsistente FM rechts. Diese Form der zusätzlichen Konsistenz nimmt stark mit dem Schwellwert für die näherungsweise Konsistenz ϵ und der Anzahl der TDOA-Messungen je Mikrofonpaar K_{ij} zu. Dies führt besonders bei der Synthese partiell konsistenter Graphen zu vielen zusätzlichen Lösungen.

Wie die ungewünschten konsistenten Kantengewichte trotzdem detektiert werden können, wird in Kap. 6.3.4 diskutiert.

6.2.3 Zusammenhang der Quellen- und Kantengewichtsanzahl

In diesem Kapitel werden die Auswirkungen der Anzahl gesuchter konsistenter Graph bzw. der Quellenanzahl Q und Anzahl der Kantengewichte K_n auf die Detektions- und Falschalarmrate betrachtet.

Ein konsistenter Graph

Für die Simulation einer Quelle $Q = 1$ wird ein voll konsistenter Graphen generiert und weitere, zufällige Messwerte hinzugefügt, womit K_n beeinflusst wird. Dies entspricht einer TDOA-Schätzung aus der GCC-PHAT, die bei niedrigem SNR oder hoher Nachhallzeit T_{60} zusätzliche, falsche TDOA-Schätzungen übergibt. Die Schätzung diser zusätzlichen TDOA-Werte ist notwendig, da die Quellenanzahl unbekannt ist und so

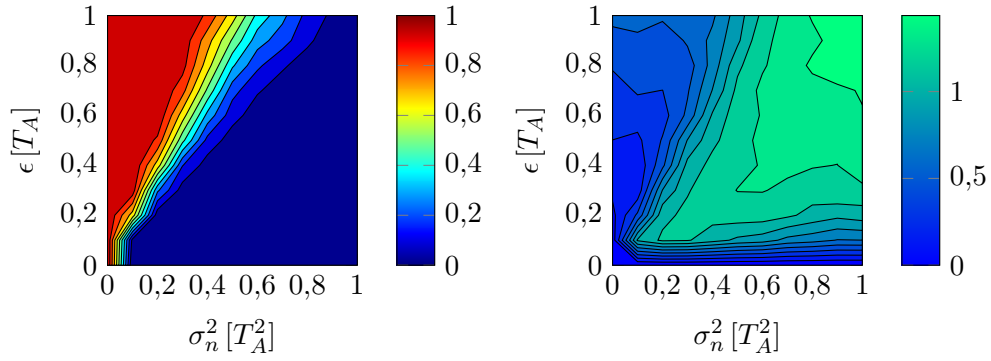
wenig richtige TDOA-Werte wie möglich verloren gehen sollen. Deshalb soll $Q \leq K_n$ gelten. In der folgenden Simulation wurde $K_n = 3$ gewählt, d.h. zwei weitere fehlerhafte Gewichte je Kante.

Da in diesem Kapitel nicht mehr die exakte Konsistenz betrachtet wird, ist der Wertebereich der Kantengewichte nicht mehr irrelevant. Für die Simulationen berechnen wir wie in Kap. 4.5 einen zweifach-zusammenhängenden Graphen und ermitteln zufällige Potenzialwerte, wie sie die Signallaufzeiten $t_{iq,0}$ darstellen, je Knoten. Die Differenz dieser Potenziale sind per Definition konsistente Kantengewichte. Die Potenziale werden über eine Gleichverteilung im Bereich $\pm\frac{w_{\max}}{2} \in \mathbb{R}$ bestimmt, so dass die Kantengewichte in einem Wertebereich $\pm w_{\max}$ liegen. Damit kann eine relative Wertedichte $\rho_w = \frac{K_n}{2w_{\max}}$ berechnet werden, die Aufschluss darauf gibt, wie weit die Kantengewichte gestreut sind. Ist ρ_w gering, dann ist die Wahrscheinlichkeit, dass sich zufällig konsistente Kombinationen ergeben, gering. Das liegt an der einhergehenden weiten Streuung der Kantengewichte in $w_n \in \left[-\frac{w_{\max}}{2}, \frac{w_{\max}}{2}\right]$, wodurch die näherungsweise Konsistenz seltener eine falsche Kantengewichtskombination als konsistent festlegt. Diese Falschalarmrate P_F, die angibt wie viele zusätzliche, partiell und näherungsweise konsistente Graphen synthetisiert werden, ist in Abb. 6.9(b) für $S = 500$ Simulationen eines Graphen mit $M = 6$ und $N = 15$ gezeigt. Es ist zu erkennen, dass für geringe Rauschleistungen σ_n^2 nahezu keine zusätzlichen Lösungen für die unterschiedlichen zyklischen Schwellwerte ϵ gefunden werden. Mit weniger als zwei zusätzlichen konsistenten Graphen ist die Falschalarmrate für $\rho_w = 0,0015\,T_A$ sehr klein. Demgegenüber zeigt Abb. 6.9(a) die Detektionsrate P_D, die angibt, wie oft der gesuchte, konsistente Graph $Q = 1$ im Durchschnitt gefunden wird. Ab einem additive weißen Rauschen mit $\sigma_n^2 > 0,6\,T_A^2$ ist es unabhängig vom Schwellwert ϵ nicht mehr möglich, den konsistenten Graphen richtig zu detektieren.

Erhöhen wir die relative Wertedichte auf $\rho_w = 0,15\,T_A$ ergeben sich die Detektions- und Falschalarmraten aus Abb. 6.10(a) bzw. Abb. 6.10(b). Es ist zu erkennen, dass SONG ebenso gut die gewünschten Lösungen detektiert. Allerdings werden viele zusätzliche, partiell konsistente Teilgraphen synthetisiert, wie Abb. 6.10(b) zeigt. Besonders für große Schwellwerte ϵ wächst die Anzahl zusätzlicher Lösungen auf über 2000 an. Die weiteren Simulationen gehen von dem größeren Wertebereich $\rho_w = 0,0015\,T_A$ aus.

Mehrere konsistente Graphen

Wenn die Quellenanzahl steigt, erhält man bei gleicher Kantengewichtsanzahl $K_n = Q = 3$ eine ähnliche Detektionsrate wie für $Q = 1$ und $K_n = 3$, wie Abb. 6.11(a) für $S = 500$ Simulationen eines Graphen mit $M = 6$ und $N = 15$ zeigt. Das ist aufgrund der Vollständigkeit von SONG zu erwarten, denn solange alle notwendigen Werte geschätzt werden, wird der zugehörige Graph synthetisiert, was besonders für



(a) Detektionsrate P_D (b) Falschalarmrate P_F

Abbildung 6.9: Detektions- und Falschalarmrate der Synthese näherungsweise konsistenter Graphen mit $M = 6$ und $N = 15$ für eine relative Wertedichte $\rho_w = 0,0015\,T_A$, $K_n = 3$ und $Q = 1$

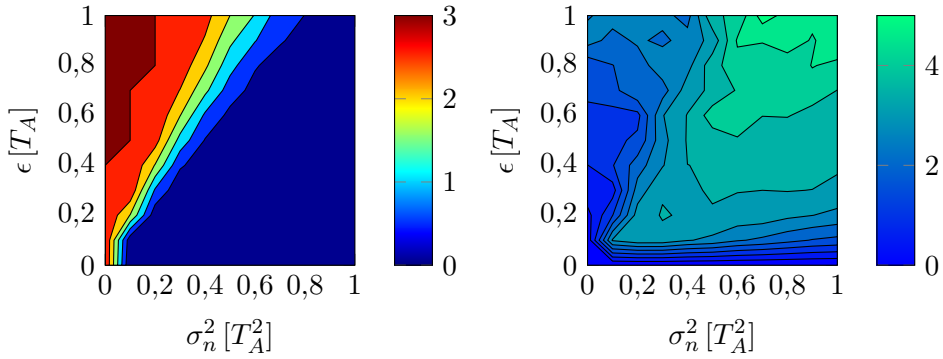
(a) Detektionsrate P_D (b) Falschalarmrate P_F

Abbildung 6.10: Detektions- und Falschalarmrate der Synthese näherungsweise konsistenter Graphen mit $M = 6$ und $N = 15$ für eine relative Wertedichte $\rho_w = 0,15\,T_A$, $K_n = 3$ und $Q = 1$

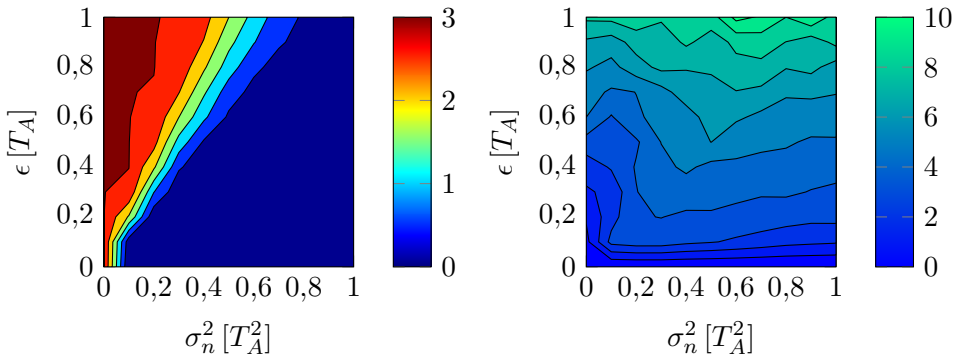
einen vollständigen Graphen gilt. Die Falschalarmrate steigt allerdings schneller an, wie Abb. 6.11(b) zeigt. Im Vergleich zu der Falschalarmrate aus Abb. 6.9(b) ist hier die Falschalarmrate mit bis zu vier zusätzlichen konsistenten Graphen bereits größer.

Werden noch mehr Kantengewichte extrahiert, so steigt die Falschalarmrate bei nahezu gleicher Detektionsrate weiter an, wie Abb. 6.11(c) und Abb. 6.11(d) für $K_n = 5$ zeigen.

Aus den hier präsentierten Ergebnissen zeigt sich, dass eine vorherige, ungefähre Abschätzung der Quellenanzahl (Ordnungsschätzung) wichtig ist. Damit reduziert sich die



(a) Detektionsrate P_D für $K_n = 3, Q = 3$ (b) Falschalarmrate P_F für $K_n = 3, Q = 3$



(c) Detektionsrate P_D für $K_n = 5, Q = 3$ (d) Falschalarmrate P_F für $K_n = 5, Q = 3$

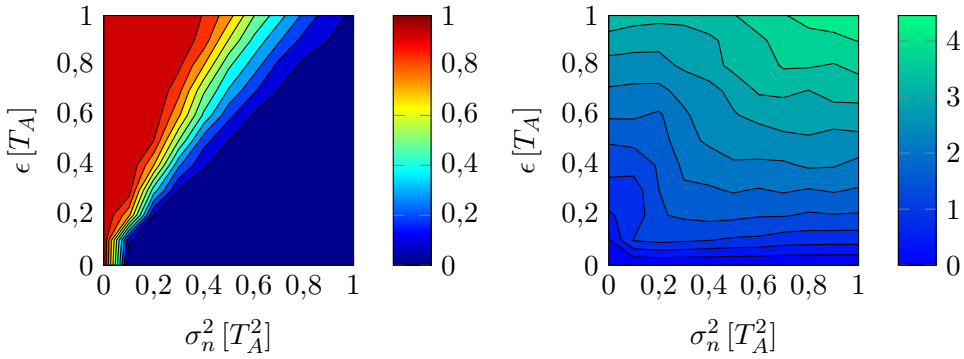
Abbildung 6.11: Detektions- und Falschalarmrate der Synthese näherungsweise konsistenter Graphen für unterschiedliche Kantengewichtsmengen K_n und unterschiedliche gewünschte Eingangsgraphen bei $\rho_w = 0,0015\,T_A$

Anzahl falscher Detektionen aufgrund von zusätzlichen, ungewünschten konsistenten Teilgraphen.

6.2.4 Auswirkungen der Bäume auf Detektions- und Falschalarmrate

Ein weiterer Unterschied ergibt sich aus den unterschiedlichen, aufspannenden Bäumen, welche maßgeblich die Topologie der FM beeinflussen. Die Detektionsrate ist für den Fall $Q = 1$ und $K_n = 3$ sowohl bei BFS als auch bei DFS gleich gut, was Abb. 6.12(a) im Vergleich zu Abb. 6.9(a) zeigt. Allerdings ergibt sich bei der Falschalarmrate ein großer Unterschied, da der DFS-Baum maximal mehr zusätzliche Lösungen generiert, wie Abb. 6.12(b) im Vergleich zu Abb. 6.9(b) zeigt.

Das liegt am zyklischen Schwellwert, der mit der Länge der FM $\sqrt{N_i}$ multipliziert



(a) Detektionsrate P_D (b) Falschalarmrate

Abbildung 6.12: Detektions- und Falschalarmrate der Synthese näherungsweise konsistenter Graphen für einen aufspannenden DFS-Baum

wird, siehe (6.16). Mit mehr Kanten je FM, wie es für DFS der Fall ist, erfüllen mehr Kantengewichtskombinationen die Konsistenzbedingung und führen zu weiteren, partiell konsistenten Graphen. Somit bestärkt sich die Verwendung eines BFS-Baumes, da dieser bereits effizienter ist.

Aus Abb. 6.12(b) und Abb. 6.9(b) ist auch ersichtlich, dass der BFS-Baum bei einem geringen Anstieg der Varianz σ_n^2 bereits nahe an den maximal möglichen Wert der Falschalarmrate gelangt. Im Vergleich dazu steigt die Falschalarmrate des DFS-Baums langsam zum maximalen Wert an.

6.2.5 Verlust von Kantengewichten

In diesem Kapitel steht die Synthese mehrerer, verrauschter Quellen im Vordergrund, deren Kantengewichte teilweise fehlen. Nach wie vor werden aber für alle N Kanten K_n Werte übergeben. Es entsteht dabei das Problem, dass fehlende Kantengewichte in den Kanten des aufspannenden Baumes nur eine teilweise Synthese des gesuchten Graphen ermöglichen. In Kap. 5.4 wurden dazu bereits für die exakte Konsistenz die zwei Verfahren des Iterieren des Wurzelknotens und die Verwendung von Kenntnissen über die Gültigkeit der Kanten eingeführt. Beispielhaft werden $Q = 3$ unterschiedliche, zweifach-zusammenhängende, konsistente Teilgraphen mit jeweils N_θ gültigen Kanten in einem vollständig zusammenhängenden Graphen generiert. Anschließend werden die Kantengewichte mit jeweils zwei zufälligen Werten auf $K_n = 5$ erweitert.

Wie Abb. 6.13 zeigt, weist das Iterationsverfahren über alle M Knoten aus Kap. 5.4.1 eine höhere Detektionsrate bei kleinen Teilgraphen $N_\theta = 3$ und eine leicht bessere Detektionsrate bei vollständigen Graphen $N_\theta = 15$ auf. Man kann zudem beobach-

ten, dass das Verfahren von Kruskal aus Kap. 5.4.2 demgegenüber eine durchschnittlich bessere Detektionsrate besitzt, da es keinen so starken Einbruch bei $N_\theta = 6$ aufweist. Diese Ergebnisse ähneln denen aus Abb. 5.8 für exakt konsistente Graphen, wonach das Iterationsverfahren aufgrund seiner BFS-Maschen der Länge drei die zweifachzusammenhängenden Graphen mit $N_\theta = 6$ nicht synthetisieren kann. Für das Verfahren von Kruskal muss aber vorausgesetzt sein, dass die Gültigkeit der Kanten ausreichend gut bestimmt wird. Für die Simulation wurde die Kantenqualität anhand der Häufigkeit der Kante in den Q generierten Teilgraphen berechnet, wie in Kap. 5.4.3.

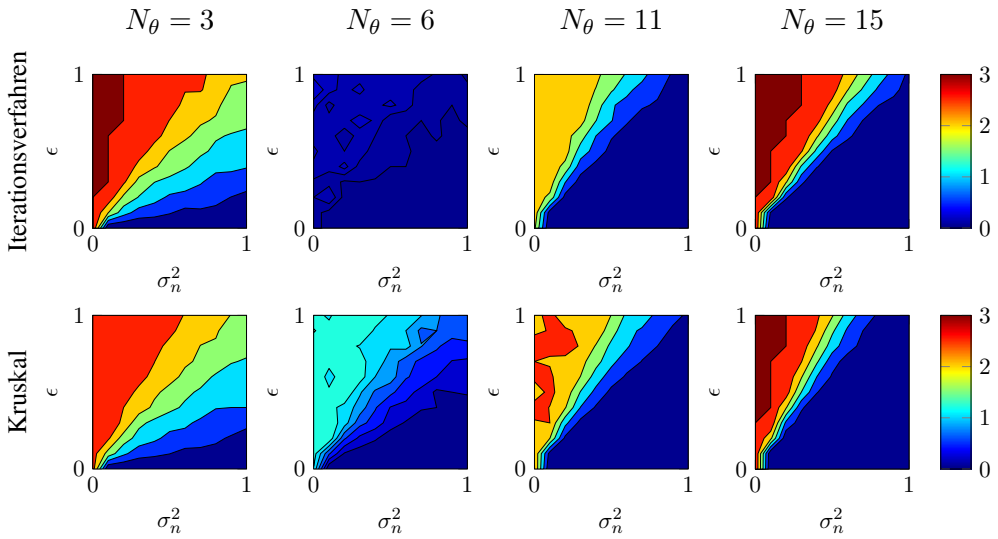


Abbildung 6.13: Detektionsraten für näherungsweise, partiell konsistente Graphen mit N_θ gültigen Kanten nach dem Iterationsverfahren und dem Verfahren nach Kruskal

Die Falschalarmraten, welche in Abb. 6.14 gezeigt sind, liegen für das Verfahren nach Kruskal für kleine, gültige Teilgraphen ($N_\theta = 3$ und 6) mit $10-16$ zusätzlichen Lösungen höher als die des Iterationsverfahren mit weniger als 10. Dafür bleiben die Falschalarmraten mit steigendem N_θ näherungsweise konstant. Im Vergleich dazu ist das Iterationsverfahren mit bis zu 50 zusätzlichen, falschen konsistenten Teilgraphen anfälliger und ineffizient.

6.3 SONG in der akustischen Lokalisierung

Nachdem die Synthese näherungsweise und partiell konsistenter Graphen im vorangegangenen Kapitel ausführlich in Simulationen diskutiert wurde, wenden wir sie in diesem Kapitel in der Lokalisierung akustischer Quellen an. Zunächst wird die Genau-

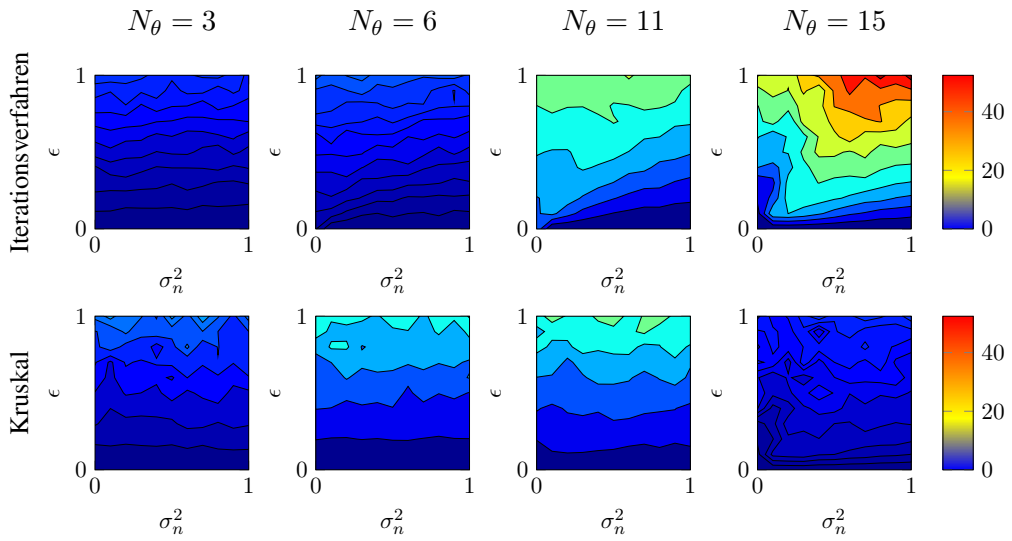


Abbildung 6.14: Falschalarmrate für mehrere partiell konsistente Graphen mit dem Iterationsverfahren und dem Verfahren nach Kruskal

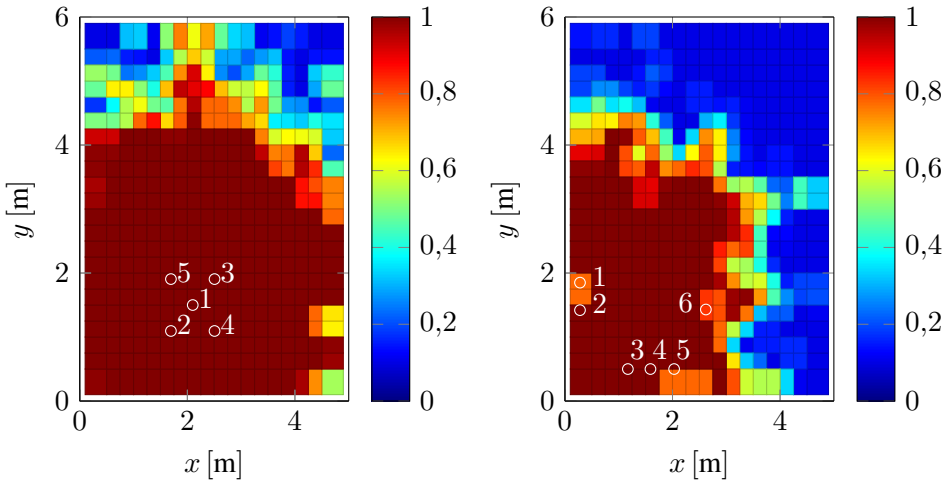
igkeit der in Kap. 2.1 besprochenen Lokalisierungsverfahren betrachtet. Anschließend wird der Einsatz im Mehrquellenszenario untersucht und schlussendlich auf eine konkrete Realisierung übergegangen.

6.3.1 Vergleich der Sensoranordnungen

Natürlich haben die geometrische Beschaffenheit des Raumes und die geometrische Anordnung der Sensoren einen erheblichen Einfluss auf die Positionsbestimmung. Wie bereits bekannt, lässt ein Sensorabstand von 50cm nur noch eine Maximalfrequenz von 342Hz zu, ohne räumliches Aliasing zu bekommen. Außerdem muss für die Mikrofonpositionen gelten, dass $\mathbf{M}$ aus (2.14) vollen Rang hat. Weiterhin muss bei der Positionierung auch darauf geachtet werden, dass die Mikrofone einen Mindestabstand von reflektierenden Gegenständen wie Wänden, Schränken oder Tischen haben, um eine klare Trennung des Direkt- und des Reflexionspfades zu erhalten. Dieser Abstand ist nach [Scheuing, 2007] mit $c_0 \cdot T_A$ gegeben.

In Abb. 6.15 wurden für zwei unterschiedliche Mikrofonarray-Geometrien die Detektionsraten in einem Raum der Größe $5 \times 6 \times 2,6\text{m}^3$ simuliert. Die Mikrofonanordnung in Abb. 6.15(a) ist ein regelmäßiges Tetraeder an dessen Ecken und in der Mitte jeweils ein Mikrofon angebracht sind. Der Abstand von der Mitte zu den Ecken betrug 70cm. In Abb. 6.15(b) ist eine unregelmäßige Anordnung eines Mikrofonarrays mit $M = 6$ Mikrofonen gegeben.

Es wurde eine Rauschquelle an den Knotenpunkten eines einheitlichen Rasters mit den Abständen 25cm simuliert. Dabei wurde die ISM-Toolbox [Lehmann, 2014] verwendet mit SNR $= 10$dB und $T_{60} = 500$ms. Die Verarbeitung der Signale erfolgte bei $f_A = 44,1$kHz für $S = 20$ Blöcke mit der Blocklänge $B = 500$ms. Die TDOA wurden als maximaler Peak in der GCC-PHAT bestimmt. Die Detektionsraten in Abb. 6.15 zeigen u.a. das räumliche Nahfeldverhalten der beiden Mikrofonarrays.



(a) Regelmäßige Tetraeder Anordnung der $M = 5$ Mikrofonen

(b) Unregelmäßige Anordnung der $M = 6$ Mikrofone

Abbildung 6.15: Vergleich der Detektionsraten im Nah- und Fernfeld für zwei unterschiedliche Mikrofonanordnungen ($\circ i$ stellt die Position des i-ten Mikrofons dar)

Die Beobachtungen können wie folgt zusammengefasst werden:

- Die Tetraederform aus Abb. 6.15(a) zeigt gute Detektionsraten selbst auf langen Distanzen, die größer als der Durchmesser sind und folglich als Fernfeld gezählt werden.
- Für regelmäßige Anordnungen ist eine weitreichende Lokalisierung möglich, was mit den Ergebnissen von [Yang u. Scheuing, 2005] übereinstimmt.
- Die unregelmäßige Anordnung aus Abb. 6.15(b) zeigt eine gute Detektionsrate im eingeschlossenen Raum zwischen den Mikrofonen. Außerhalb ist ein schneller Abfall zu erkennen.
- Für eine große Entfernung zwischen der Quelle und den Mikrofonen (Fernfeld) ist die Signalenergie an den Mikrofonen sehr gering, so dass keine vernünftige TDOA-Schätzung erfolgen kann.

- Für große Mikrofonabstände ist zudem die maximal darstellbare Frequenz aufgrund von räumlichem Aliasing geringer und führt zu Problemen in der Lokalisierbarkeit.

6.3.2 Cramer-Rao-Schranke der Einzelquellenlokalisierung

Nachdem die Sensorgeometrien betrachtet wurden, konzentrieren wir uns in diesem Kapitel auf den Lokalisierungsfehler. Dafür wird die Annahme getroffen, dass die TDOA-Schätzfehler klein und Bias-frei sind. Dementsprechend kann man über die Fisher-Informationsmatrix $\mathbf{F}$ die Cramer-Rao-Schranke (CRS) als eine untere Schranke für die Varianz der Positionsfehler berechnen [Huang u. a., 2004]. Dafür wird die Wahrscheinlichkeitsdichtefunktion (WDF) $p(\underline{\delta}|\underline{p})$ für die gemessenen Abstandsdifferenz $\delta_{ij} = \tau_{ij} c_0$ in Abhängigkeit von der Quellposition $\underline{p}$ nach (2.7) betrachtet. Die CRS ist dann

$$\mathbf{C}_{\mathrm{CR}} = \mathbf{F}^{-1} \quad \text{mit} \quad \mathbf{F} = \mathrm{E}\left[\left(\bigtriangledown_{\underline{p}} \ln p(\underline{\delta}|\underline{p})\right)\left(\bigtriangledown_{\underline{p}} \ln p(\underline{\delta}|\underline{p})\right)^T\right] \tag{6.20}$$

[Köhler, 2005]. Dabei ist $\mathrm{E}(X) = \int_{-\infty}^{\infty} x \cdot f_X(x) dx$ der Erwartungswert einer Zufallsvariablen X mit der zugehörigen WDF $f_X(x)$. Der Gradientenvektor ist $\bigtriangledown_{\underline{p}} = [\frac{\partial}{\partial p_1}, \ldots, \frac{\partial}{\partial p_D}]^T$. Für den Fall der unabhängigen, normalverteilten Differenzschätzung, wie in (2.32) angenommen, vereinfacht sich die CRS zu

$$\mathbf{C}_{\mathrm{CR}} = \left(\bigtriangledown_{\underline{p}}^T (\underline{\delta} - \underline{d}(\underline{p}))\right)^T \mathbf{C}^{-1} \left(\bigtriangledown_{\underline{p}}^T (\underline{\delta} - \underline{d}(\underline{p}))\right) \tag{6.21}$$

mit

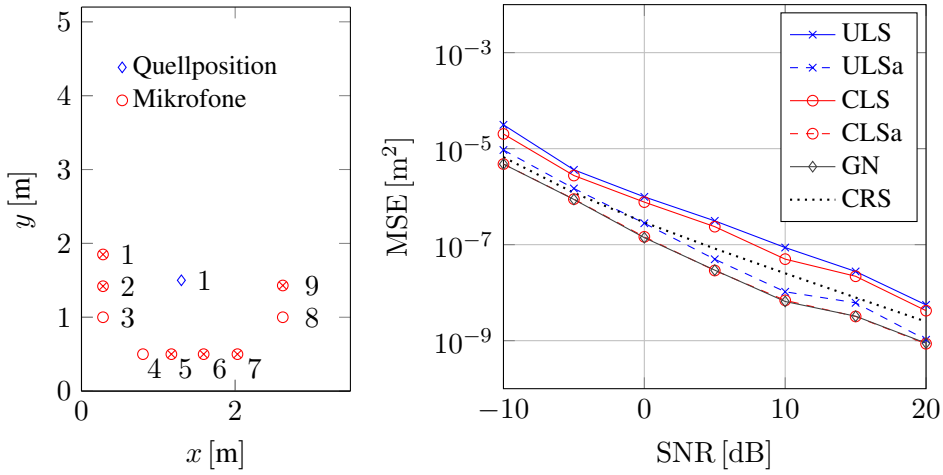
$$\bigtriangledown_{\underline{p}}^T (\underline{\delta} - \underline{d}(\underline{p})) = \mathbf{J} = \begin{bmatrix} (\underline{u}_1 - \underline{u}_2)^T \\ \vdots \\ (\underline{u}_{M-1} - \underline{u}_M)^T \end{bmatrix}, \tag{6.22}$$

den normierten Einheitsvektoren $\underline{u}_i = (\underline{p}_q - \underline{m}_i)/(||\underline{p}_q - \underline{m}_i||)$ und den Mikrofonpositionen $\underline{m}_i,\ i = 1, \ldots, M$, siehe auch (2.44) [Chan u. Ho, 1994a; So u. a., 2008]. Da die TDOA aber nicht ganz Bias-frei geschätzt werden können, was in den Simulationen bestätig wurde und aufgrund der nichtlinearen Problemstellung der Quellenlokalisierung kann die CRS, wenn überhaupt, nur asymptotisch erreicht werden [Huang u. a., 2004] und dient in dieser Arbeit als grobe Verifikation der Ergebnisse.

6.3.3 Genauigkeit der Verfahren zur Einzelquellenlokalisierung

Um zunächst ein gutes Einzelquellenokalisierungsverfahren auszuwählen, wurden die idealen TDOA $\underline{t} = [t_{12}, \ldots, t_{(M-1)M}]^T$ mit einem additiven, unabhängigen und identischen normalverteilten Rauschen ρ_{ij} gestört. Als Fehlermaß wird der mittlere quadratische Fehler (mean squared error, MSE) als euklidischer Abstand der Positionsschätzung

zur Quellposition bestimmt. Für die Genauigkeitsberechnung der Verfahren wurden nur die Mikrofone $1, 2, 5, 6, 7$ und 9 aus Abb. 6.16(a) eingesetzt und daraus die idealen TDOA ermittelt um Fehler der TDOA-Schätzung auszuschließen. Diese TDOA wurden anschließend mit dem Rauschen ρ_{ij} der Varianz aus (6.10) überlagert. Die Blocklänge betrug hierbei $B = 500$ms. Über $S = 1000$ Versuchen gemittelt, erhält man die MSE aus Abb. 6.16(b).

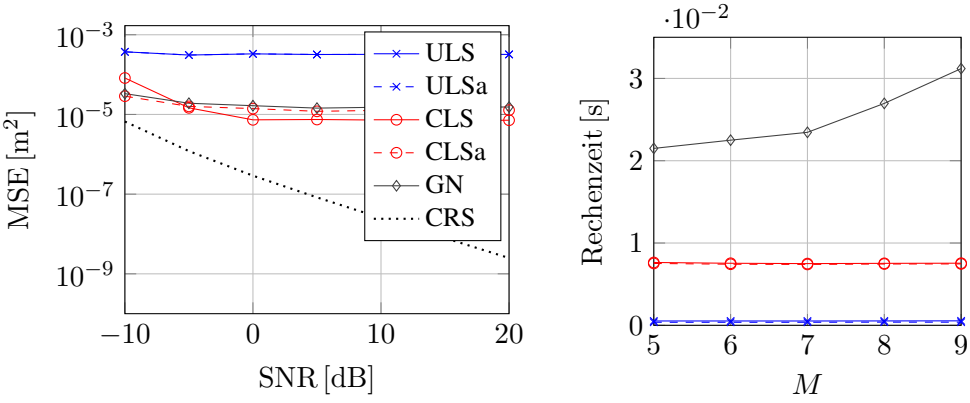


(a) Anordnung der Mikrofone (b) Mittlerer quadratischer Fehler der Positionsschätzung

Abbildung 6.16: Vergleich der Einzelquellenlokalisierungsverfahren für eine feste Sensorgeometrie in Bezug auf Genauigkeit bei Störung der idealen TDOA- Werte

Es zeigt sich, dass eine ausreichend gute Genauigkeit mit einem mittleren quadratischen Fehler kleiner als $0,001\text{m}^2$ für ein SNR bis -10dB erreicht wird. Des Weiteren ist zu erkennen, dass das GN-Verfahren unter Verwendung aller TDOA-Werte und der schrittweisen Annäherung an das Optimumz6666666665 wie erwartet am besten abschneidet. Interessanterweise ist das CLS-Verfahren mit der Abbildung aller TDOA-Werte auf den Satz nichtredundanter TDOA (CLSa) aus Kap. 2.1.4 genauso exakt. Das nächste exakte Verfahren ist das ULS-Verfahren mit der Abbildung der TDOA-Werte (ULSa). Zur Verifikation der Ergebnisse ist die CRS aus Kap. 6.3.2 eingezeichnet. Diese kann teilweise unterschritten werden, da sie nur für erwartungstreue Schätzer die minimale Schranke darstellt. Bei der nichtlinearen Problemstellung der Positionsschätzung existiert ein solcher Schätzer nicht [Yang u. Scheuing, 2005]. Deshalb ist sie nur als Orientierung zu sehen und nicht als kleinste, mögliche Varianz. Wie erwartet kann zuletzt festgestellt werden, dass die Abbildung der vollen TDOA-Menge auf die sphärische TDOA-Menge eine Verbesserung in der Genauigkeit der Positionsschätzung zeigt. Dies gilt sowohl für das ULS als auch das CLS-Verfahren.

Betrachtet man den realistischeren Fall eines Rauschens an den Mikrofonsignalen ρ_i nach (2.57), anstelle eines Rauschens direkt an den TDOA wie in Abb. 6.16(b), so ergibt sich ein geänderter Verlauf über das SNR. Es wurde dabei die Impulsantwort mit der ISM von [Lehmann u. Johansson, 2008] und einer kurzen Nachhallzeit von $T_{60} = 100$ms berechnet, um die Raumeinflüsse zu minimieren. Bei einer Blocklänge von $B = 500$ms, $f_A = 48$kHz und quadratischer Interpolation für die TDOA-Schätzung erhält man die Positionsgenauigkeiten aus Abb. 6.17(a). Die Ergebnisse weisen eine deutliche Verschlechterung auf, was an der Signaldämpfung entlang des Ausbreitungspfades und der zeitdiskreten Verarbeitung der GCC-PHAT liegt. Es ist zudem eine asymptotische Sättigung ab 0dB zu beobachten. Diese Sättigung liegt am mittelwertbehafteten Quantisierungsfehler, der durch die quadratische Interpolation nur teilweise verringert werden kann. Dieser Bias ist durch die relative Position der Quelle zur Position der Mikrofone gegeben. Aufgrund dieses Bias wird die CRS von keinem Verfahren erreicht.



(a) Mittlerer quadratischer Fehler der Positionsschätzung bei umgerechnetem SNR von den Mikrofonsignalen zu den TDOA nach (6.10)

(b) Rechenzeit der unterschiedlichen Einzelquellen Lokalisierungsverfahren

Abbildung 6.17: Positionsgenauigkeit bei simulierten Quellsignalen und additivem Rauschen an den Mikrofonen sowie die Rechenzeit der einzelnen Lokalisierungsverfahren (ohne TDOA-Schätzung)

Es zeigt sich eine ähnliche Abstufung in der Genauigkeit der Verfahren wie in Abb. 6.16(b). Dementsprechend zählen das CLSa und GN-Verfahren zu den exaktesten Verfahren. Überraschenderweise zeigt die Abbildung der TDOA von der vollen auf die sphärische Menge beim ULS-Verfahren keine Verbesserung und beim CLS-Verfahren sogar eine minimale Verschlechterung.

Neben der Genauigkeit der Einzelquellenlokalisierungsverfahren ist für unsere Betrachtung noch die Rechenzeit von Bedeutung. In Abb. 6.17(b) sind die gemittelten Zeiten

über eine ansteigende Anzahl von Mikrofonen dargestellt. Hierbei handelt es sich um die $M = 9$ Mikrofone aus Abb. 6.16(a). Wie zu erwarten braucht das GN-Verfahren durch seine Iterationen am längsten und die zwei Verfahren CLS und CLSa durch die zweistufige Verarbeitung mit dem Lagrange-Operator über 5ms. Am effizientesten sind die zwei Verfahren ULS und ULSa, da diese mit einer Matrixinvertierung auskommen. Im Folgenden wird das ULS-Verfahren verwendet, da es schnell ist und eine ausreichend gute Genauigkeit für unseren Anwendungsfall hat.

6.3.4 Simulation von Mehrquellen

Um das ULS-Verfahren in einem Mehrquellenszenario anzuwenden, benutzen wir folglich den SONG-Algorithmus, um die Mehrdeutigkeiten zu reduzieren. Hierbei werden sowohl Ergebnisse aus einem einzelnen BFS-Baum, dem Iterationsverfahren mit M BFS-Bäumen und dem Baum nach Kruskal und Prim betrachtet. Die Simulationen und Messungen werden bei einer Abtastfrequenz von $44,1$kHz durchgeführt. Die Blocklänge beträgt 8192 Samples (~ 200ms) und wurde mit einem Rechteckfenster gewichtet. Der Phasentransformationsfilter und die quadratische Interpolation nach (6.7) wurden angewendet und es erfolgte eine simple Pausenerkennung über die Summe der Beträge der Abtastwerte nach (6.15) in einem Signalblock.

Qualitätsmass der TDOA [Scheuing, 2007] führte ein Qualitätsmass ein, das die einzelnen TDOA-Werte anhand ihrer relativen Position mit einer Glockenkurve gewichtet. Das Zentrum der Glockenkurve bestimmt sich dabei aus den Peaks der Autokorrelation der Mikrofonsignale, welche ein Indiz für den Direktpfad und die Reflexionspfade geben. Daraus entsteht eine Direktpfaderkennung, welche den TDOA ein Verlässlichkeitsmass zuweist. Diese als *Rasterbedingung* bekannte Methode impliziert den erhöhten Rechenaufwand der Autokorrelationsberechnung.

Deshalb wird hier ein anderes Qualitätsmass betrachtet, welches für die Suche nach dem aufspannenden Baum verwendet werden kann. Besonders für das Verfahren von Kruskal ist die Verlässlichkeit einer Kante von Bedeutung, da hiermit die möglichen falschen Kombinationen der FM reduziert wird, wie in Kap. 5.3 diskutiert. Hierfür ist es sinnvoll, die Signalenergie der einzelnen TDOA-Maxima als Grundlage zu nehmen. Die Qualität q_n einer Kante $e_n = (i, j)$ ist somit die Summe der TDOA Maxima je Kante:

$$q_n = \sum_{k=1}^{K_n} |r_{ij}[\hat{n}_k]|. \tag{6.23}$$

Lokalisierung von Rauschquellen

Für die Simulation und Messung in diesem Kapitel wird das Array aus Abb. 6.18 gewählt. Zunächst werden $Q = 3$ Quellen simuliert, die jeweils unterschiedliche Rauschsignale wiedergeben bei einer langen Nachhallzeit von $T_{60} = 500\text{ms}$ und einem realistischen SNR von 20dB. Nach der TDOA-Schätzung wird SONG ausgeführt und für jeden konsistenten TDOA-Graphen die Positionen mit dem ULS-Verfahren bestimmt. Dabei konnten nur die konsistenten TDOA-Graphen für die Positionsschätzung verwendet werden, die einen Knoten mit mindestens vier nichtredundanten TDOA-Werten besitzen, damit $\mathbf{P}$ aus (2.17) für den 3-dimensionalen Fall invertierbar ist.

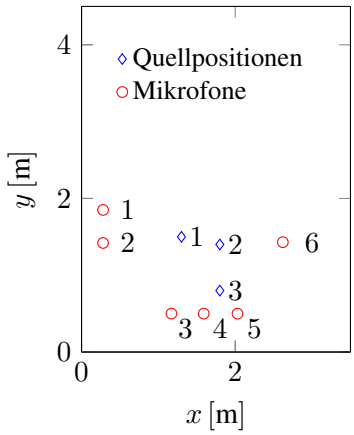


Abbildung 6.18: Anordnung des Mehrquellenszenarios mit drei Quellen und sechs Mikrofonen

Gemittelt über 20 Blöcke bekommt man die Detektionsraten in Tab. 6.4. Für die Detektion gilt, dass der euklidische 3-dimensionale Abstand zwischen der realen Quellposition und einer geschätzten Position in einem Signalblock kleiner als 5cm sein muss. Dann gilt die Quelle als detektiert und fließt in die Detektionsrate P_D ein. Es gilt hier $P_D \leq Q = 3$. Die Falschalarmrate P_F ist die durchschnittliche Anzahl von Positionsschätzungen außerhalb dieses Radius und $\hat{K}$ ist die durchschnittliche Anzahl von Positionsschätzungen je Block, da nicht alle partiell konsistenter Graphen für eine Lokalisierung mit ULS dienen. Es gilt $P_D + P_F \leq \hat{K}$, weil jede Quelle nur einmal je Block als detektiert gezählt wird, es aber mehr Positionsschätzungen innerhalb des Radius geben kann.

In Tab. 6.4 ist gut zu erkennen, dass zusätzliche, partiell konsistente Graphen entstehen. Besonders das Verfahren mit M BFS-Bäumen generiert durch seinen mehrfachen Aufruf viele partiell konsistente Graphen, die nicht zu den gewünschten TDOA-Kombinationen führen. Der zyklische Schwellwerte war $\epsilon = 0,3\,T_A$. Eine Verkleinerung des Schwellwertes führt zu einer kleineren P_D, weil dann auch die gewünschten

	ein BFS Baum	M BFS Bäume	Kruskal und Prim
P_D	$1,8$	$3,0$	$2,6$
P_F	$0,2$	$15,0$	$3,7$
$\hat{K}$	$2,2$	$26,6$	$7,4$
Zeit	$0,086$s	$0,136$s	$0,093$s

Tabelle 6.4: Detektions- und Falschalarmraten der unterschiedlichen Syntheseverfahren bei der simultanen Lokalisierung von drei akustischen Rauschquellen, $T_{60} = 500$ms, SNR$= 20$dB, $K_n = 3$ und $\epsilon = 0,3\,T_A$

TDOA-Kombinationen nicht mehr detektiert werden können. Das Verfahren nach Kruskal und Prim stellt einen guten Kompromiss dar, bei einer akzeptablen Rechenzeit.

Die Quellen werden bei dieser Simulation nicht immer erkannt, weil ihre zugehörigen TDOA nicht gut geschätzt werden, wie in Abb. 6.19 verdeutlicht wird. In der GCC-PHAT zwischen Mikrofon 3 und 4 werden die TDOA-Werte zu jeder Quelle △ durch jeweils ein Maxima in der GCC-PHAT □ richtig geschätzt. Im Gegensatz dazu kommt es zwischen dem Mikrofon 1 und 2 aufgrund der geometrischen Anordnung dazu, dass das Maximum der Quelle 2 von dem Maximum der Quelle 1 verdeckt wird. Deshalb kann in diesem Beispiel Quelle 2 nicht genau genug detektiert werden und fällt bei der näherungsweisen Konsistenzbedingung raus.

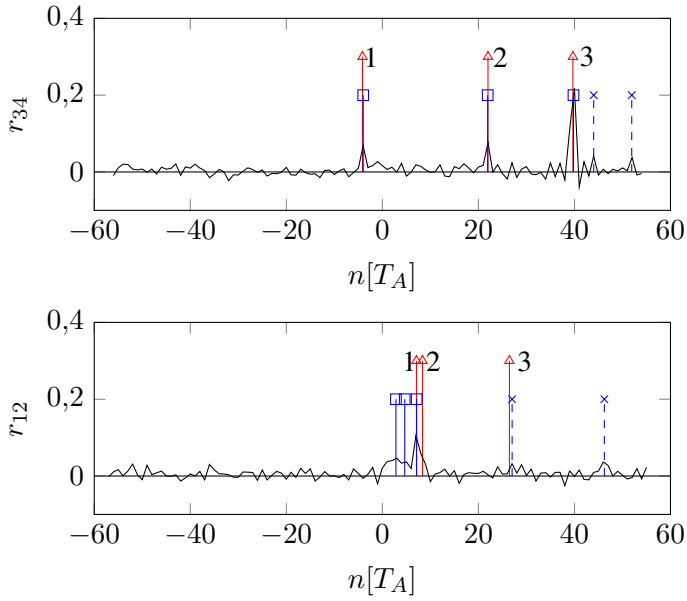


Abbildung 6.19: Zwei GCC-PHAT mit den richtigen △ und geschätzten (□, ×) TDOA für die Simulation aus Tab. 6.4

Da in der Synthese mit einem vollständig zusammenhängenden Graphen der Sensor 1 meistens als Wurzelknoten gewählt wird, wird das Mikrofonpaar $(1, 2)$ im aufspannenden BFS-Baum enthalten sein. Dieses Problem wird durch die M-fache Anwendung ausgeglichen, weil die restlichen TDOA der Quelle 2 soweit richtig detektiert wurden und τ_{12} dann im Kobaum liegt und nur eine FM stört. Somit kann der konsistente Graph zu Quelle 2 in dem Fall ausreichend gut synthetisiert werden.

Zusätzlich ist das Maxima von Quelle 3 insgesamt zu gering in r_{12}, um unter den drei größten Maxima in Abb. 6.19 zu sein, was P_D weiterhin reduziert. Erhöht man die Anzahl der Kantengewichte $K_n = 5$, so werden weitere Maxima geschätzt, die in Abb. 6.19 als $\times$ markiert sind, und die Detektionsrate steigt, wie Tab. 6.5 zeigt. Das geht zu Lasten von P_F, weil sich mehr zufällige Kombinationen ergeben, die zu verfälschten Positionsschätzungen führen.

	ein BFS Baum	M BFS Bäume	Kruskal und Prim
P_D	$2,10$	$3,00$	$2,65$
P_F	$1,95$	$43,50$	$12,20$
$\hat{K}$	$4,65$	$62,65$	$20,00$
Zeit	$0,089$s	$0,213$s	$0,129$s

Tabelle 6.5: P_D und P_F der unterschiedlichen Syntheseverfahren bei der simultanen Lokalisierung von drei akustischen Rauschquellen, $T_{60} = 500$ms, SNR$=$ 20dB, $K_n = 5$ und $\epsilon = 0,3\,T_A$

Eine Erhöhung des Schwellwertes ϵ führt dazu, dass zu viele falsche TDOA-Werte akzeptiert werden. Damit wird für die ULS-Lokalisierung eine schlechte sphärische TDOA-Menge ausgewählt und viele geschätzte Positionen liegen weiter als 5cm neben der richtigen Position.

In Abb. 6.20 sind in jeder Zeile ein idealer TDOA-Graph von einer Quelle und der zugehörige, partiell konsistente Graph, der von SONG für unterschiedliche ϵ in der Simulation synthetisiert wurde gezeigt. Zufälligerweise stimmt in dieser Simulation der Wurzelknoten für die BFS-Suche mit dem Wurzelknoten der sphärischen TDOA-Menge für die ULS-Lokalisierung überein. Für Quelle 3 werden die TDOA relativ gut geschätzt, was an der Nähe zu den Mikrofonen $3, 4, 5$ und 6 liegt. Somit ist der große Abstand zu Mikrofon 1 kein großes Problem. Zu Mikrofon 2 ist aber keine gute Schätzung mehr möglich, da das Signal auf beiden Pfaden zu 1 und 2 gleich stark gedämpft wird und somit Quelle 3 in der GCC-PHAT r_{12} nur ein kleines Maxima hat. Trotzdem werden die anderen TDOA genau genug geschätzt, so dass der synthetisierte konsistente Graph zu Quelle 3 einen kleinen Fehler in der Positionsschätzung von $0,3$cm hat. Hier führt auch eine Erhöhung des zyklischen Schwellwerts ϵ zu keiner Verbesserung. Für Quelle 1, die für $\epsilon = 0,3\,T_A$ eine gute Positionsschätzung mit einem Fehler von nur $1,9$cm hat, bewirkt die Erhöhung auf $\epsilon = 0,5\,T_A$, dass eine schlechte FM $\{v_1, v_4, v_5\}$ hinzukommt

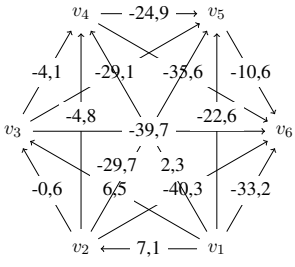
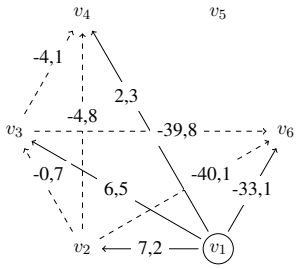
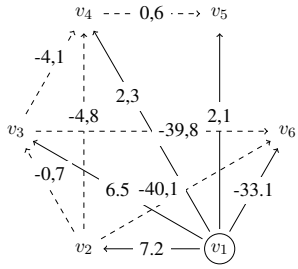
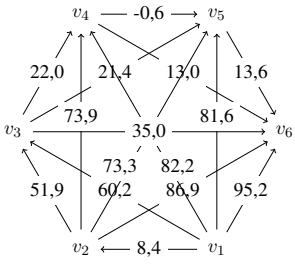
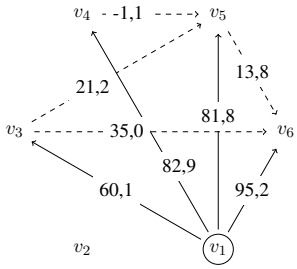
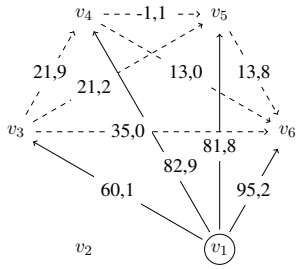
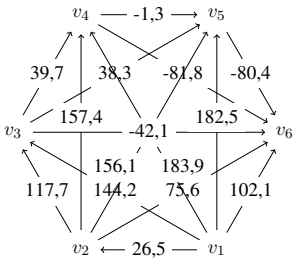
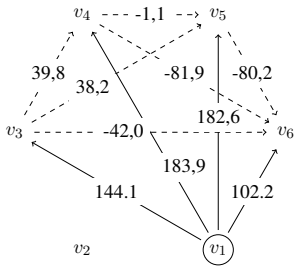
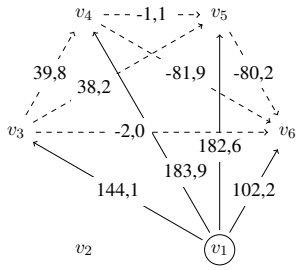
Ideale TDOA	SONG mit $\epsilon = 0,3\,T_A$	SONG mit $\epsilon = 0,5\,T_A$
		
$\underline{p}_1 = \begin{bmatrix} 1,300\text{m} & 1,500\text{m} \end{bmatrix}^T$	$\hat{\underline{p}}_1 = \begin{bmatrix} 1,295\text{m} & 1,492\text{m} \end{bmatrix}^T$ $\lvert \underline{p}_1 - \hat{\underline{p}}_1 \rvert = 1,9\text{cm}$	$\hat{\underline{p}}_1 = \begin{bmatrix} 1,696\text{m} & 1,923\text{m} \end{bmatrix}^T$ $\lvert \underline{p}_1 - \hat{\underline{p}}_1 \rvert = 56,8\text{cm}$
		
$\underline{p}_2 = \begin{bmatrix} 1,800\text{m} & 1,400\text{m} \end{bmatrix}^T$	$\hat{\underline{p}}_2 = \begin{bmatrix} 1,793\text{m} & 1,397\text{m} \end{bmatrix}^T$ $\lvert \underline{p}_2 - \hat{\underline{p}}_2 \rvert = 10,3\text{cm}$	$\hat{\underline{p}}_2 = \begin{bmatrix} 1,793\text{m} & 1,398\text{m} \end{bmatrix}^T$ $\lvert \underline{p}_2 - \hat{\underline{p}}_2 \rvert = 10,3\text{cm}$
		
$\underline{p}_3 = \begin{bmatrix} 1,800\text{m} & 0,800\text{m} \end{bmatrix}^T$	$\hat{\underline{p}}_3 = \begin{bmatrix} 1,800\text{m} & 0,801\text{m} \end{bmatrix}^T$ $\lvert \underline{p}_3 - \hat{\underline{p}}_3 \rvert = 0,3\text{cm}$	$\hat{\underline{p}}_3 = \begin{bmatrix} 1,800\text{m} & 0,801\text{m} \end{bmatrix}^T$ $\lvert \underline{p}_3 - \hat{\underline{p}}_3 \rvert = 0,3\text{cm}$

Abbildung 6.20: Theoretische TDOA-Graphen und aus der Simulation am ähnlichsten synthetisierte, partiell konsistente Graphen. Die TDOA-Werte sind in $[T_A]$ gegeben. Der umrundete Knoten entspricht dem Wurzelknoten der Synthese und der sphärischen TDOA-Menge, $K_n = 3$

und das Gewicht der Kante (v_1, v_5) zu einer Verschlechterung der Positionsschätzung führt. Deshalb ist es besser, den Schwellwert ϵ klein zu halten. Quelle 2 wird allgemein schlecht geschätzt, was nur durch eine Erhöhung von K_n behoben werden kann, womit eine Erhöhung von P_F einhergeht.

Dieselben Probleme, wie das Anfügen schlechter konsistenter Tripel und Fehlen der

richtigen TDOA, treten auch bei dem Verfahren von Kruskal und Prim auf. Allerdings sind durch die gezielte Auswahl des aufspannenden Baumes diese Einflüsse reduziert.

Lokalisierung von Sprachquellen

Im Vergleich zu den vorangegangenen Rauschsignalen an den Quellen stellen die Sprachsignale eine größere Schwierigkeit dar. Deshalb wurden unterschiedliche Kombinationen von Schwellwerten ϵ und Kantengewichtsgrößen K_n simuliert. Mit den drei Syntheseverfahren, basierend auf den unterschiedlichen aufspannenden Bäumen, bekommt man für Sprachquellen mit dem gleichen Aufbau wie in Abb. 6.18 für die Rauschquellen die Ergebnisse aus Tab. 6.6.

K_n	3	3	3	5	5	5	8	8	8
ϵ	$0,3\,T_A$	$0,7\,T_A$	$1,2\,T_A$	$0,3\,T_A$	$0,7\,T_A$	$1,2\,T_A$	$0,3\,T_A$	$0,7\,T_A$	$1,2\,T_A$
ein BFS-Baum									
P_D	0,25	0,30	0,35	0,35	0,40	0,50	0,35	0,65	1,45
P_F	0,00	0,65	1,40	0,35	7,90	29,00	4,40	54,65	306,25
$\hat{K}$	0,25	1,05	2,20	0,75	9,10	31,10	5,25	58,30	354,05
Zeit	0,075s	0,075s	0,078s	0,076s	0,091s	0,129s	0,086s	0,170s	0,951s
M BFS-Bäume									
P_D	0,65	0,85	0,95	0,70	1,15	1,60	1,25	2,55	2,80
P_F	1,25	10,60	44,35	7,25	72,85	335,10	56,65	665,40	3534,95
$\hat{K}$	4,90	22,50	84,45	12,60	94,10	405,60	68,35	710,60	3771,55
Zeit	0,086s	0,116s	0,253s	0,106s	0,255s	1,185s	0,218s	2,420s	34,731s
Kruskal und Prim									
P_D	0,65	0,85	0,85	0,85	1,30	1,45	1,30	2,40	2,70
P_F	0,35	3,05	9,65	3,20	34,40	124,15	45,05	440,05	1738,80
$\hat{K}$	1,35	5,95	17,90	5,45	46,75	155,30	54,60	489,85	1933,80
Zeit	0,079s	0,087s	0,106s	0,091s	0,184s	0,584s	2,187s	35,868s	247,622s

Tabelle 6.6: P_D und P_F der unterschiedlichen Syntheseverfahren bei unterschiedlichem K und ϵ für die simultane Lokalisierung von drei Sprachquellen, $T_{60} = 500$ms, SNR$= 20$dB

Es zeigt sich, dass mit einem einzigen BFS-Baum nur unzureichend genau lokalisiert werden kann, wobei die Falschalarmrate immer noch akzeptabel bleibt. Für die Verfahren mit M BFS-Bäumen und dem Baum nach Kruskal und Prim können wir mit der Kombination $K_n = 8$ und $\epsilon = 0,7\,T_A$ gute Detektionsraten von $P_D > 2$ für die drei Quellen erhalten. Auffällig ist bei diesen zwei Verfahren die hohe Falschalarmrate, die ohne Weiteres nicht reduziert werden kann. Auch wenn das Verfahren mit dem aufspannenden Baum nach Kruskal und Prim insgesamt weniger konsistente Graphen synthetisiert, ist die Rechenzeit höher als beim Verfahren mit M BFS-Bäumen. Das zeigt, dass der Baum von Kruskal und Prim lange FM generiert, die viele Kantengewichtskombinationen überprüfen.

Tabelle 6.7 zeigt die bestmöglichen TDOA-Schätzungen $\underline{\hat{w}}_q$ für die Sprachquellen in einem Block. Die Spalten $\underline{g}_q$ stellen die idealen TDOA-Werte dar und $\underline{\hat{g}}_q$ sind die synthetisierten TDOA-Graphen. Dabei wurden die Parameter $K_n = 8$ und $\epsilon = 1,0\,T_A$ sowie das Verfahren mit einem BFS-Baum verwendet.

i	j	$\underline{g}_1$	$\underline{\hat{w}}_1$	$\underline{\hat{g}}_1$	$\underline{g}_2$	$\underline{\hat{w}}_2$	$\underline{\hat{g}}_2$	$\underline{g}_3$	$\underline{\hat{w}}_3$	$\underline{\hat{g}}_3$
1	2	7,10	6,87	6,87	8,37	6,87	25,77	26,46	25,77	51,04
1	3	6,47	6,39	6,39	60,24	67,14	103,33	144,20	103,33	103,33
1	4	2,33	2,27	2,27	82,24	85,82	85,82	183,88	208,91	104,16
1	5	−22,60	−21,98	−21,98	81,64	80,02	86,84	182,54	188,62	188,62
1	6	−33,24	−33,82	−33,82	95,23	69,88	69,88	102,11	69,88	69,88
2	3	−0,63	−0,92	−0,92	51,88	15,14	–	117,73	124,99	–
2	4	−4,77	−4,92	−4,92	73,87	53,98	–	157,41	158,76	53,98
2	5	−29,70	−29,92	−29,92	73,28	62,02	62,02	156,07	62,02	–
2	6	−40,34	−40,08	−40,08	86,86	7,64	–	75,65	7,64	–
3	4	−4,14	−4,04	−4,04	22,00	8,76	–	39,68	40,77	−0,28
3	5	−29,07	−29,02	−29,02	21,40	29,05	−16,30	38,34	37,38	–
3	6	−39,71	−39,94	−39,94	34,98	12,96	–	−42,09	−39,94	–
4	5	−24,93	−25,30	−25,30	−0,60	0,55	0,55	−1,34	−2,03	–
4	6	−35,57	−35,61	−35,61	12,99	15,15	–	−81,77	−82,82	−35,61
5	6	−10,64	−10,11	–	13,58	32,00	−16,10	−80,43	−119,94	−119,94
	x	1,30m	1,28m	1,28m	1,80m	1,64m	1,63m	1,80m	1,76m	1,80m
	y	1,50m	1,49m	1,49m	1,40m	1,37m	1,48m	0,80m	0,98m	0,99m
$\lvert\underline{p}_q - \underline{\hat{p}}_q\rvert$			0,02m	0,02m		0,15m	0,18m		0,19m	0,19m

Tabelle 6.7: TDOA-Werte der jeweiligen Quellen $\underline{g}_q$, die bestmöglichen geschätzten TDOA-Werte $\underline{\hat{w}}_q$ und der jeweilige synthetisierte Graph $\underline{\hat{g}}_q$. Die TDOA-Werte sind in $[T_A]$ gegeben ($K_n = 8, \epsilon = 1,0\,T_A$, ein BFS-Baum)

Aus der Tab. 6.7 wird ersichtlich, dass bereits die TDOA-Schätzungen schlecht sind, wie in den Spalten $\underline{\hat{w}}_q$ angedeutet ist. Hier sind die nächstgelegenen TDOA-Schätzungen zu den richtigen TDOA-Werten selektiert worden. Es zeigt sich, dass bereits diese bestmögliche TDOA-Kombinationen mit dem ULS-Verfahren zu Positionen führen, die bei Quelle 2 und 3 mehr als 10cm entfernt liegen.

Man kann sehen, dass gerade diejenigen Sprachsignale besonders schlechte TDOA-Schätzungen haben, die überhaupt nur mit einem hohen Schwellwerten ϵ detektiert werden können. Dies hat natürlich eine hohe Anzahl an Falschdetektionen zur Folge, die im nächsten Abschnitt behandelt werden.

Reduzierung der Falschalarmrate

Die Konsistenzbedingung stellt eine notwendige aber nicht hinreichende Bedingung dar, wie die zusätzlichen konsistenten Lösungen aus Kap. 6.2.2 zeigen. Deshalb wird hier ein Nachverarbeitungsschritt eingeführt, um die ungewünschten, konsistenten Lö-

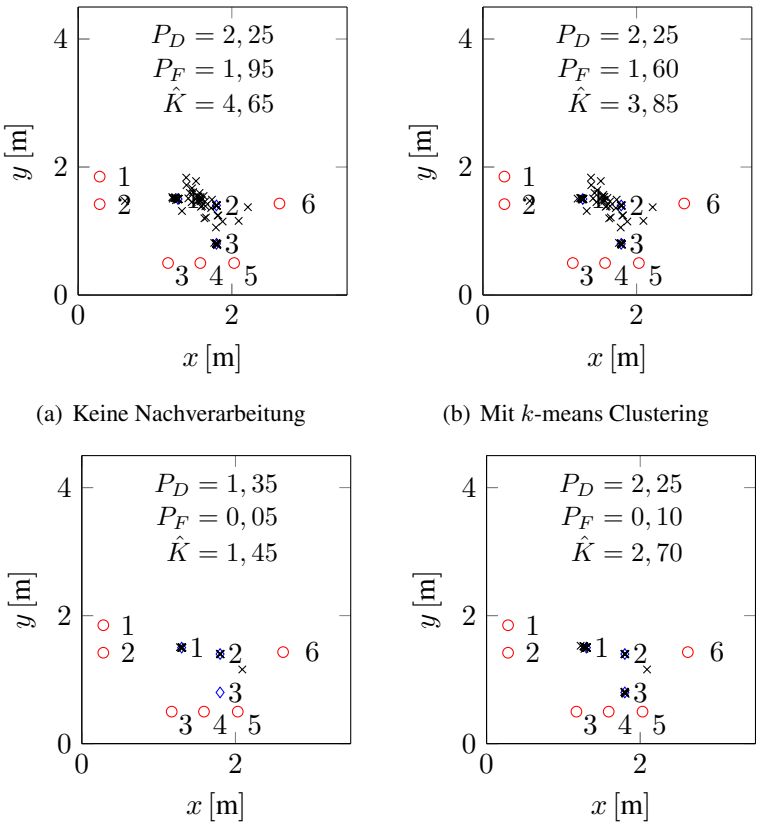
sungen zu detektieren. Dazu werden drei unterschiedliche Ansätze betrachtet und diese mittels Simulationen verglichen.

Selektion gültiger TDOA-Kombinationen In Kap. 2.3.2 wurde die TDOA-basierte Schallgeschwindigkeitsschätzung eingeführt, die in der Mehrquellenlokalisierung für die Selektion gültiger TDOA-Kombinationen eingesetzt wurde [Hu u. Yang, 2010]. Dementsprechend berechnet man für jede konsistente TDOA-Kombination in einem Nachbearbeitungsschritt die Schallgeschwindigkeit anhand des Nahfeldmodells und überprüft nach (2.59) aus Kap. 2.3.2 auf Gültigkeit. Ist dabei die berechnete Abweichung der Schallgeschwindigkeit geringer als der Schwellwert e_c, so wird der konsistente TDOA-Graph einer gültigen Quelle zugeordnet. Dieses Verfahren wurde bereits in [Annibale u. a., 2013b] angewendet.

Entsprechend der Schallgeschwindigkeit wenden wir auch das Kriterium des residualen Fehlers aus (2.64) auf eine konsistente TDOA-Kombination an und evaluieren diese anhand des Schwellwertes e_t auf Gültigkeit.

Clustering der Positionsschätzungen Wie in Abb. 6.21(a) gezeigt ist, liegen viele Positionsschätzungen nahe um die richtigen Positionen verteilt. Daher bietet es sich an, die Positionsschätzungen mit dem k-means-Algorithmus zu gruppieren [Lloyd, 1982]. Dabei werden alle Positionen in k Gruppen aufgeteilt, so dass jede Positionsschätzung einen minimalen quadratischen Abstand zu einem der k Zentren aufweist. Da in unserem Mehrquellenlokalisierungsverfahren die Anzahl der Quellen als unbekannt vorausgesetzt wird, wird das erweiterte, adaptive k-means-Verfahren verwendet [Hu u. a., 2009; Lee u. Choi, 2010]. Dieses Verfahren inkrementiert die Anzahl der Gruppen k so lange, bis alle Positionen näher als ein gegebener Abstand a an einem Zentrum liegen.

Vergleich der Selektionsverfahren Um die Effektivität der genannten Verfahren zur Reduzierung der Falschalarmrate zu vergleichen, wurde das Setup aus Abb. 6.18 verwendet mit Rauschquellen und $T_{60} = 500\text{ms}$. Als maximaler Abstand des adaptiven k-means-Algorithmus wurde der maximale Abstand einer gültigen Positionsschätzung zu $a = 5\text{cm}$ gewählt. Der optimale Schwellwert für das Selektionsverfahren mittels Schallgeschwindigkeitsschätzung ergab sich mit $e_c = 11\frac{\text{m}}{\text{s}}$ und für den residualen Fehlers bei $e_t = 0,4\mu\text{s}$. Angewendet wurde hierbei das Syntheseverfahren mit einem aufspannenden BFS-Baum, einem zyklische Schwellwert von $\epsilon = 0,3T_A$ und $K_n = 5$. In Abb. 6.21 sind alle Positionsschätzungen über 20 Signalblöcke der Länge 186ms bei $f_A = 44,1\text{kHz}$ dargestellt, sowie die mittlere Detektions- und Falschalarmrate. Diese beziehen sich auf einen Abstand von 5cm im $D = 2$ dimensionalen Fall und nicht $D = 3$ wie in den bisherigen Auswertungen.



(a) Keine Nachverarbeitung

(b) Mit k-means Clustering

(c) Mit Schallgeschwindigkeitsschätzung

(d) Mit residualem Fehler

Abbildung 6.21: Ergebnisse der Mehrquellenlokalisierung mit unterschiedlichen Nachverarbeitungsverfahren

Abbildung 6.21(a) zeigt, dass in dieser Simulation maximal $2,25$ von 3 Quellen simultan detektiert werden können. Zudem erhält man durchschnittlich $1,95$ zusätzliche falsche Positionen. Wenn eines der Nachverarbeitungsverfahren hingezunommen wird, erhält man mit dem k-means-Clustering die Ergebnisse in Abb. 6.21(b), mit der Schätzung der Schallgeschwindigkeit die Ergebnisse aus Abb. 6.21(c) und mit dem Verfahren des residualen Fehlers Abb. 6.21(d). Es zeigt sich, dass das adaptive k-means-Verfahren die mittlere Anzahl konsistenter Lösungen $\hat{K}$ und somit die Falschalarmrate leicht reduziert und dabei die maximale Anzahl möglicher richtiger Positionsschätzungen beibehält. Demgegenüber ist die Nachverarbeitung mit der Schallgeschwindigkeitsschätzung restriktiver, da wenige Positionen geschätzt werden, die aber dafür fast ausschließlich richtig sind. Abbildung 6.21(c) zeigt, dass die Quelle 3 dabei nie gefunden wird. Als besten Nachverarbeitungsschritt ist das Verfahren des residualen Fehlers zu nennen, da

es eine maximale Anzahl möglicher Positionsschätzungen und eine sehr kleine Falschalarmrate hat.

6.3.5 Messung in realen Räumen

Nachdem die möglichen Störeinflüsse in den vorangegangenen Kapiteln genau analysiert wurden, wird der SONG-Algorithmus hier in einem Lokalisierungsszenario mit drei Sprachquellen in einem Büro der Größe $3,5 \times 4,5 \times 3\text{m}^3$ wie in Abb. 6.21 und einer Nachhallzeit von $T_{60} \approx 450\text{ms}$ angewendet, siehe auch [Kreißig u. Yang, 2013]. Die Lautsprecher und Mikrofone werden an den Positionen in Tab. 6.8 aufgestellt und es werden drei unterschiedliche Sprachsignale mit einer Länge von 6s abgespielt und analysiert.

	$\underline{p}_1$	$\underline{p}_2$	$\underline{p}_3$	$\underline{m}_1$	$\underline{m}_2$	$\underline{m}_3$	$\underline{m}_4$	$\underline{m}_5$	$\underline{m}_6$
$x\,[\text{m}]$	$1,30$	$1,80$	$1,80$	$0,28$	$0,28$	$1,17$	$1,59$	$2,03$	$2,62$
$y\,[\text{m}]$	$1,50$	$1,40$	$0,80$	$1,85$	$1,42$	$0,50$	$0,50$	$0,50$	$1,43$
$z\,[\text{m}]$	$1,30$	$1,30$	$1,30$	$1,30$	$1,30$	$1,50$	$1,50$	$1,50$	$1,50$

Tabelle 6.8: Positionen der Mikrofone und Lautsprecher für die Echtzeitlokalisierung mit SONG

Für eine schnelle Verarbeitung wurde die Abtastrate auf $f_A = 16\text{kHz}$ reduziert und die quadratische Interpolation auf der GCC-PHAT angewendet. Außerdem wurde der Detektionsradius auf 10cm erhöht und eine Blocklänge von 512ms gewählt. Zunächst wurde der aufspannender Baum nach Kruskal und Prim verwendet und mit einer unterschiedlichen Anzahl von TDOA-Kandidaten K_n gestartet. Mit $\epsilon = 0,4\,T_A$ ergeben sich die Detektionsraten $P_{D,q}$ und durchschnittliche Laufzeiten je Signalblock aus Tab. 6.9.

K_n	$\hat{K}$	$P_{D,1}$	$P_{D,2}$	$P_{D,3}$	t_{tot} [s]	t_{SONG} [s]	M_{cc}	p_E	$p_{\bar{E}}$
4	$49,17$	$0,50$	$0,67$	$0,67$	$0,099$	$0,018$	$64,5$	$0,009$	$0,155$
5	$110,00$	$0,83$	$0,67$	$1,00$	$0,119$	$0,062$	$107,7$	$0,005$	$0,161$
6	$224,67$	$1,00$	$0,83$	$1,00$	$0,198$	$0,127$	$151,8$	$0,005$	$0,159$
7	$646,00$	$1,00$	$1,00$	$1,00$	$1,269$	$1,162$	$290,2$	$0,003$	$0,162$
8	$1026,33$	$1,00$	$1,00$	$1,00$	$3,311$	$3,163$	$332,0$	$0,003$	$0,162$

Tabelle 6.9: Ergebnisse der Lokalisierung von drei Sprachquellen in einem Büro mit SONG-Algorithmus

Bereits für $K_n = 7$ können alle Quellen zu 100% erkannt werden. Dies geschieht aber bei einer hohen Rechenzeit, die nicht echtzeitfähig ist und einer hohen Anzahl zusätzlicher Lösungen. In dieser Messung ist keine Nachverarbeitung durchgeführt worden.

Für Anwendungen, die nicht zeitkritisch sind und bei denen eine hohe Detektionsrate wichtig ist, kann dieses Setup gut eingesetzt werden. Für einen effizienteren Einsatz sollte $K_n = 6$ TDOA-Kandidaten je Mikrofonpaar gewählt werden, die mit einer Rechenzeit $< 0,2$s unter der Dauer eines Signalblocks von $0,512$s liegen. Es wird zudem ersichtlich, dass neben der TDOA-Schätzung und der Positionsbestimmung je konsistentem Graph der SONG-Algorithmus den größten Anteil der Rechenzeit benötigt. Zusätzlich ist in Tab. 6.9 für die Analyse des CC-Graph die Anzahl konsistenter FM $M_{cc} = \sum_{i=1}^{N_{\mathrm{FM}}} \tilde{K}_i$ gezeigt und davon abgeleitet die relative Anzahl von FM-Paaren, die kompatibel $p_E = |E(G_{cc})|/N_{cc}$ oder in Konflikt $p_{\bar{E}} = |\bar{E}(G_{cc})|/N_{cc}$ sind. Man erkennt, dass nur wenige konsistente FM überhaupt kompatibel sind, weshalb der SONG-Algorithmus für diese Anwendung auch effizient anwendbar ist.

6.3.6 Vergleich zu SRP-PHAT

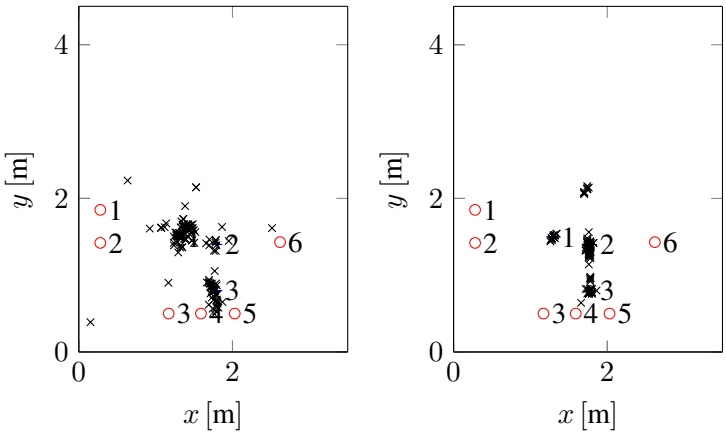
Zuletzt vergleichen wir den in dieser Arbeit vorgestellten Ansatz zur Lokalisierung simultaner akustischer Quellen mit dem SRP-PHAT-Verfahren, das in Kap. 2.3.5 eingeführt wurde. Dabei wurde eine Schwelle von $P_{\mathrm{thresh}} = 0,5 \cdot \max\left(P_{\mathrm{SRP-PHAT}}\right)$ als Kriterium für eine Quellendetektion definiert. Die örtliche Schrittweite des SRP-PHAT-Verfahrens wurde zu 2cm gewählt. Beide Verfahren werden für den $D = 2$ dimensionalen Fall ausgewertet. Für das SONG-Verfahren wurden der aufspannende Baum von Kruskal und Prim gewählt, $K_n = 7$, $\epsilon = 0,4\,T_A$ gewählt, sowie bei der TDOA-Schätzung der GCC-Phat-Filter und die quadratische Interpolation angewendet. Damit würden sich P_D und P_F sowie die Laufzeiten aus Tab. 6.9 ergeben. Da P_F sehr hoch ist, schalten wir die Überprüfung des residualen Fehlers aus Kap. 6.3.4 als Nachverarbeitung hinzu und bekommen mit $e_t = 6,3\mu$s die Werte aus Tab. 6.10.

	P_D	P_F	Zeit	$\bar{r}_{\mathrm{det}}$
SONG	$2,67$	$21,67$	$1,519$s	$5,1$cm
SRP-PHAT	$2,83$	$17,83$	$2,855$s	$4,1$cm

Tabelle 6.10: Vergleich von P_D und P_F sowie der Rechenzeit und durchschnittliche Positionsfehler der richtigen Positionsschätzungen für die Lokalisierung mit SONG und SRP-PHAT

Es ist zu erkennen, dass sowohl das SONG- als auch das SRP-PHAT-Verfahren gute Detektionsraten haben, wobei der SONG-Algorithmus durch die Nachverarbeitung einen erkennbaren Verlust gegenüber der 100% von P_D aus Tab. 6.9 zeigt. SONG ist dabei fast doppelt so schnell wie SRP-PHAT und bewirkt mit der Nachverarbeitung eine erhebliche Reduzierung von P_F von $646,00$ aus Tab. 6.9 auf $21,67$ falschen Positionsschätzungen. Die Rechenzeit für SONG kann – wie in Tab. 6.9 gezeigt – weiter reduziert werden, wenn ein Verlust von richtigen Positionsschätzungen möglich ist. Für

SRP-PHAT hingegen ist die Rechenzeit nahezu fest, da immer das gesamte Raumraster abgesucht werden muss.



(a) SONG (b) SRP-PHAT

Abbildung 6.22: Die Positionsschätzungen des SONG- und des SRP-PHAT-Verfahrens für eine Messung mit drei Sprachsignalen in einem Büro

Die Positionsschätzungen der beiden Verfahren sind in Abb. 6.22 abgebildet. Es zeigt sich, dass die TDOA-basierten Positionsschätzungen mit dem SONG-Algorithmus eine größere Streuung aufweisen als die Schätzungen des SRP-PHAT-Verfahrens Das bestätigt auch der durchschnittliche euklidische Abstand $\bar{r}_{\mathrm{det}}$ einer Detektion zu der nächstgelegenen Quelle aus Tab. 6.10. Hier ist SRP-PHAT besser als der TDOA-basierte Ansatz.

Kapitel 7

Zusammenfassung und Ausblick

Zusammenfassung

In der vorliegenden Arbeit wurde die Synthese konsistenter Graphen (synthesis of consistent graphs, SONG) betrachtet, wie sie in der Laufzeitdifferenz (TDOA)-basierten Lokalisierung akustischer Quellen vorkommen. Ein konsistenter Graph ist eine Menge von Knoten, Kanten und Kantengewichten, wobei die Summe der Kantengewichte entlang aller Maschen im Graph Null ergibt. In der Mehrquellenlokalisierung können die TDOA als Kantengewichte zwischen den einzelnen Mikrofonen (Knoten) abstrahiert werden. Stammen die TDOA von der gleichen Quelle, so ist der zugehörige Graph konsistent. Diese Eigenschaft macht man sich zu Nutze, um die Mehrdeutigkeiten in der Zuordnung der TDOA zwischen den Mikrofonpaaren aufzulösen und robust mehrere Quellen simultan zu lokalisieren. Dazu misst man mehrere TDOA je Mikrofonpaar und generiert daraus einen Graphen mit mehreren Kantengewichten. Anschließend synthetisiert man mit SONG alle konsistenten Graphen und führt für jeden konsistenten Graphen eine Lokalisierung durch.

In dieser Arbeit wurde gezeigt, dass mittels eines beliebigen aufspannenden Baumes ein minimaler Satz von fundamentalen Maschen gefunden werden kann, der repräsentativ für alle Maschen in einem Graph ist. Berechnet man für diese fundamentalen Maschen alle konsistenten Kantengewichtskombinationen, so bilden diese den Ausgangspunkt für SONG.

Der SONG-Algorithmus kann danach zum einen das Backtracking-Verfahren anwenden, um effizient alle voll konsistenten Graphen zu synthetisieren, die auf allen Kanten ein Kantengewicht haben. Da in den meisten Anwendungen wie der Lokalisierung manche Kantengewichte aufgrund von Schätzfehlern fehlen können, ist es notwendig, auch partiell konsistente Graphen zu synthetisieren, die nicht auf allen Kanten ein Kantengewicht haben. Deshalb bietet SONG auch die Möglichkeit, alle partiell konsistenten Graphen mittels des neuen Suchverfahrens CCGsearch zu synthetisieren. Dieses basiert auf dem Kompatibilitäts-Konflikt-Graph (Compatibility-Conflict-Graph, CC-Graph), der eine neue Form von Graph ist und drei Zustände zwischen seinen Kno-

ten zulässt: offen, kompatibel und in Konflikt. Die Knoten repräsentieren dabei die konsistenten fundamentalen Maschen, die entweder keine gemeinsamen Kanten haben (offen), gemeinsame Kanten mit gleichen Kantengewichten (kompatibel) oder mit unterschiedlichen Kantengewichten haben (Konflikt). CCGsearch führt dabei eine vollständige Suche nach allen kompatiblen Knotenmengen durch.

Sowohl das Backtracking-Verfahren als auch CCGsearch wurden in dieser Arbeit bezüglich ihres Detektionsverhaltens und Effizienz analysiert und in Simulationen verifiziert. Dabei zeigte sich eine starke Abhängigkeit von der Topologie des aufspannenden Baumes und den zu synthetisierenden konsistenten Graphen. Wird für die Suche nach dem aufspannenden Baum eine Breitensuche (BFS) angewendet, so sind die fundamentalen Maschen kurz und es können schnell alle konsistenten Kantengewichtskombinationen gefunden werden. Es hat sich aber gezeigt, dass Graphentopologien mit großen Maschen nicht von kurzen fundamentalen Maschen synthetisiert werden können, weshalb die Suche von Kruskal und Prim für den aufspannenden Baum betrachtet wurde. Dieses Verfahren kann die Qualität einer Kante mit einbeziehen, die bei einer Anwendung die Verlässlichkeit der gemessenen Werte sein kann. Bei dieser Suche kann es vorkommen, dass die fundamentalen Maschen lange sind und die Rechenzeit stark zunimmt. Für die Anwendung in der akustischen Lokalisierung konnte gezeigt werden, dass ein BFS-Baum ausreichend ist und dass der Wurzelknoten bei der Suche nach dem aufspannenden Baum einen hohen Knotengrad haben soll.

Für die Lokalisierung wurden unterschiedliche Ansätze betrachtet und deren Lokalisierungsgenauigkeiten verglichen. Es zeigte sich, dass sowohl die Verfahren basierend auf der sphärischen TDOA-Menge, als auch der vollen TDOA-Menge eine ausreichend gute Genauigkeit aufweisen und dass der größte Einflussfaktor für die Positionsgenauigkeit die TDOA-Schätzung aus der Kreuzkorrelation der Mikrofonsignale ist.

Nach der Erweiterung von SONG auf die näherungsweise Konsistenz, damit verrauschte TDOA-Werte verarbeitet werden können, wurde der Einsatz von SONG in der Mehrquellenlokalisierung zunächst anhand von Simulationen und anschließend anhand von realen Messungen gezeigt. Aufgrund der hohen Anzahl an zusätzlichen Positionsschätzungen durch konsistente Teilgraphen wurden in dieser Arbeit noch unterschiedliche Nachverarbeitungsschritte verglichen, um die Anzahl falscher Detektionen zu verringern. Hierbei hat sich der residuale Fehler als ein robuster Selektionsmechanismus bewährt. Die Genauigkeit, Detektions- und Falschalarmrate, sowie die Laufzeit wurden anschließend in Simulationen und Messungen ermittelt und mit dem Beamforming-Verfahren SRP-PHAT verglichen. Dabei erreichen beide Lokalisierungsverfahren gleich gute Erkennungsraten, wobei die Lokalisierung mit SONG eine geringere Laufzeit benötigt und die Ergebnisse mehr streuen.

Ausblick

In [Oualil u. a., 2013] werden die TDOA in der Mehrquellenlokalisierung als eine Mischung von Gaußverteilungen angenommen. Anschließend wird diese Verteilungsfunktion auf den euklidichen Raum projeziert, um damit die Regionen mit einer hohen Wahrscheinlichkeit für eine Quelle zu bestimmen. Dieser Ansatz verwendet keine harte Entscheidung bei der TDOA-Schätzung und kann in Kombination mit dem SONG Algorithmus zu einer effizienteren, verbesserten Lokalisierung führen.

Neben der Lokalisierung akustischer Quellen in geschlossenen Räumen ist auch die Lokalisierung von Funksensoren in einem verteilten Netzwerk ein interessanter Anwendungsfall für SONG [Ji u. Zha, 2003]. Hierbei werden räumlich weit verteilte Knoten betrachtet, die sich untereinander lokalisieren, um somit Informationen über den betrachteten Bereich zu sammeln.

Für den CCGsearch-Ansatz, der in der Komplexität schnell wächst, ist auch der effiziente Ansatz der Dynamischen Programmierung denkbar [Cormen u. a., 2009]. Dieser bricht ein gegebenes Optimierungsproblem auf dessen Teilprobleme herab, wie es die konsistenten, fundamentalen Maschen für SONG darstellen und kombiniert diese Teillösungen. Dabei handelt es sich um ein Optimierungsverfahren, das die beste Lösung zu einem Problem findet und nicht alle möglichen Lösungen liefert. Dieser Algorithmus kann aber für die Synthese konsistenter Graphen sequentiell angewendet werden und somit im Gegensatz zu SONG mit CCGsearch die besten Lösungen liefern. Dieser Ansatz kann für manche Anwendungen besser sein.

Literaturverzeichnis

[Akkoyunlu 1973] AKKOYUNLU, E A.: The Enumeration of Maximal Cliques of Large Graphs. In: *SIAM Journal on Computing* 2 (1973), Nr. 1, S. 1–6. http://dx.doi.org/10.1137/0202001. – DOI 10.1137/0202001. – ISSN 0097–5397

[Annibale u. a. 2013a] ANNIBALE, Paolo ; FILOS, Jason ; NAYLOR, Patrick A. ; RABENSTEIN, Rudolf: TDOA-based Speed of Sound Estimation for Air Temperature and Geometric Inference. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 21 (2013), Nr. 2, S. 234–246. http://dx.doi.org/10.1109/TASL.2012.2217130. – DOI 10.1109/TASL.2012.2217130. – ISSN 1558–7916

[Annibale u. Rabenstein 2010] ANNIBALE, Paolo ; RABENSTEIN, Rudolf: Accuracy of Time-Difference-of-Arrival Based Source Localization Algorithms under Temperature Variations. In: *International Symposium on Communications, Control and Signal Processing* (2010), S. 1–4. http://dx.doi.org/10.1109/ISCCSP.2010.5463320. – DOI 10.1109/ISCCSP.2010.5463320. ISBN 9781424462872

[Annibale u. Rabenstein 2013] ANNIBALE, Paolo ; RABENSTEIN, Rudolf: Closed-form estimation of the speed of propagating waves from time measurements. In: *Multidimensional Systems and Signal Processing* (2013), S. 361–378. http://dx.doi.org/10.1007/s11045-013-0231-x. – DOI 10.1007/s11045–013–0231–x. – ISSN 0923–6082

[Annibale u. a. 2013b] ANNIBALE, Paolo ; RABENSTEIN, Rudolf ; KREISSIG, Martin ; YANG, Bin: Joint consistent graph synthesis and speed of sound estimation for acoustic localization in multi-source reverberant environments. In: *International Workshop on Multidimensional Systems* (2013), S. 1–6. ISBN 978–3–8007–3543–3

[Antoniou u. Lu 2007] ANTONIOU, Andreas ; LU, Wu-Sheng: *Practical Optimization: Algorithms and Engineering Applications*. Springer, 2007. – ISBN 0387711066

[Balabanian u. Bickart 1969] BALABANIAN, Norman ; BICKART, Theodore A.: *Electrical Network Theory*. John Wiley & Sons, 1969. – ISBN 9780471045762

[Beck u. a. 2008] BECK, Amir ; STOICA, Petre ; LI, Jian: Exact and Approximate Solutions of Source Localization Problems. In: *IEEE Transactions on Signal Processing* 56 (2008), Nr. 5, S. 1770–1778. http://dx.doi.org/10.1109/TSP.2007.909342. – DOI 10.1109/TSP.2007.909342. – ISSN 1053–587X

[Boehme u. a. 1998] BOEHME, H.-J. ; BRAUMANN, U.-D. ; BRAKENSIEK, A. ; CORRADINI, A. ; KRABBES, M. ; GROSS, H.: User localisation for visually-based human-machine-interaction. In: *IEEE International Conference on Automatic Face and Gesture Recognition* (1998), S. 486–491. http://dx.doi.org/10.1109/AFGR.1998.670995. – DOI 10.1109/AFGR.1998.670995

[Boyd u. Vandenberghe 2004] BOYD, Stephen ; VANDENBERGHE, Lieven: *Convex Optimization*. Cambridge University Press, 2004. – ISBN 0521833787

[Brandstein u. Silverman 1997] BRANDSTEIN, Michael S. ; SILVERMAN, Harvey F.: A practical methodology for speech source localization with microphone arrays. In: *Computer Speech & Language* 11 (1997), Nr. 2, S. 91–126. http://dx.doi.org/10.1006/csla.1996.0024. – DOI 10.1006/csla.1996.0024. – ISSN 0885–2308

[Bron u. Kerbosch 1973] BRON, Coen ; KERBOSCH, Joep: Algorithm 457: Finding All Cliques of an Undirected Graph. In: *Communications of the ACM* 16 (1973), Nr. 9, S. 575–577. http://dx.doi.org/10.1145/362342.362367. – DOI 10.1145/362342.362367. – ISSN 00010782

[Bronstein u. Semendjajew 2008] BRONSTEIN, I.N. ; SEMENDJAJEW, K.A.: *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 2008. – ISBN 3817120176

[Carevic 2007] CAREVIC, D.: Automatic Estimation of Multiple Target Positions and Velocities Using Passive TDOA Measurements of Transients. In: *IEEE Transactions on Signal Processing* 55 (2007), Nr. 2, S. 424–436. http://dx.doi.org/10.1109/TSP.2006.885745. – DOI 10.1109/TSP.2006.885745. – ISSN 1053–587X

[Carter 1987] CARTER, G C.: Coherence and Time Delay Estimation. In: *Proceedings of the IEEE* 75 (1987), Nr. 2, S. 236–255. http://dx.doi.org/10.1109/PROC.1987.13723. – DOI 10.1109/PROC.1987.13723. – ISSN 0018–9219

[Carter u. a. 1973] CARTER, G. C. ; NUTTALL, A. H. ; CABLE, P. G.: The smoothed coherence function. In: *Proceedings of the IEEE* 61 (1973), S. 1497–1498. http://dx.doi.org/10.1109/PROC.1973.9300. – DOI 10.1109/PROC.1973.9300

[Chan u. Ho 1994a] CHAN, Y T. ; HO, C K.: A Simple and Efficient Estimator for Hyperbolic Location. In: *IEEE Transactions on Signal Processing* 42 (1994), Nr. 8, S. 1905–1915. http://dx.doi.org/10.1109/78.301830. – DOI 10.1109/78.301830. – ISSN 1053–587X

[Chan u. Ho 1994b] CHAN, Y.T. ; HO, K.C.: An efficient closed-form localization solution from time difference of arrival measurements. In: *IEEE International Conference on Acoustics, Speech and Signal Processing* (1994), S. 393–396. http://dx.doi.org/10.1109/ICASSP.1994.389638. – DOI 10.1109/ICASSP.1994.389638. ISBN 0–7803–1775–0

[Chen u. a. 2007] CHEN, Xiaojie ; YUANCHUN, Shi ; JIANG, Wenfeng: Speaker Tracking and Identifying Based on Indoor Localization System and Microphone Array. In: *International Conference on Advanced Information Networking and Applications Workshops* 2 (2007), S. 347–352. http://dx.doi.org/10.1109/AINAW.2007.341. – DOI 10.1109/AINAW.2007.341

[Cormen u. a. 2009] CORMEN, Thomas H. ; LEISERSON, Charles E. ; RIVEST, Ronald L. ; STEIN, Clifford: *Introduction To Algorithms*. 3. The MIT Press, 2009. – ISBN 9780262033848

[Dechter 2003] DECHTER, Rina: *Constraint processing*. Morgan Kaufmann, 2003. – ISBN 1558608907

[Dibiase u. a. 2001] DIBIASE, Joseph H. ; SILVERMAN, Harvey F. ; BRANDSTEIN, Michael S.: Robust localization in reverberant rooms. In: BRANDSTEIN, Michael S. (Hrsg.) ; WARD, Darren (Hrsg.): *Microphone Arrays*. Springer Verlag, 2001. – ISBN 978–3–540–41953–2, S. 157–180

[Diestel 2006] DIESTEL, R: *Graphentheorie (3.Auflage)*. Springer Verlag, 2006. – ISBN 3540213910

[Dijkstra 1959] DIJKSTRA, E W.: A note on two problems in connexion with graphs. In: *Numerische Mathematik* 1 (1959), Nr. 1, S. 269–271. http://dx.doi.org/10.1007/BF01386390. – DOI 10.1007/BF01386390. – ISSN 0029–599X

[Do u. a. 2007] DO, Hoang ; SILVERMAN, Harvey F. ; YU, Ying: A Real-Time SRP-PHAT Source Location Implementation using Stochastic Region Contraction(SRC) on a Large-Aperture Microphone Array. In: *IEEE International Conference on Acoustics, Speech and Signal Processing* (2007), S. 121–124. http://dx.doi.org/10.1109/ICASSP.2007.366631. – DOI 10.1109/ICASSP.2007.366631. ISBN 1–4244–0727–3

[Drews 1995] DREWS, Martin: Time Delay Estimation for Microphone Array Speech Enhancement Systems. In: *European Conference on Speech Communication and Technology* (1995), S. 2013–2016

[Eisenbrand u. a. 2005] EISENBRAND, Friedrich ; FUNKE, Stefan ; KARRENBAUER, Andreas ; REICHEL, Joachim ; SCHÖMER, Elmar: Packing a trunk: now with a twist! In: *Proceedings of the ACM symposium on Solid and Physical Modeling* (2005), S. 197–206. http://dx.doi.org/10.1145/1060244.1060266. – DOI 10.1145/1060244.1060266. ISBN 1595930159

[Foy 1976] FOY, Wade H.: Position-Location Solutions by Taylor-Series Estimation. In: *IEEE Transactions on Aerospace and Electronic Systems* 12 (1976), Nr. 2, S. 187–194. http://dx.doi.org/10.1109/TAES.1976.308294. – DOI 10.1109/TAES.1976.308294. – ISSN 0018–9251

[Friedlander 1987] FRIEDLANDER, Benjamin: A Passive Localization Algorithm and Its Accuracy Analysis. In: *IEEE Journal Of Oceanic Engineering* (1987), S. 234–245. http://dx.doi.org/10.1109/JOE.1987.1145216. – DOI 10.1109/JOE.1987.1145216. – ISSN 0364–9059

[Gerhard 2006] GERHARD, Doblinger: Localization and Tracking of Acoustical Sources. Version: 2006. http://dx.doi.org/10.1007/3-540-33213-8_4. In: HÄNSLER, Eberhard (Hrsg.) ; SCHMIDT, Gerhard (Hrsg.): *Topics in Acoustic Echo and Noise Control*. Springer Berlin Heidelberg, 2006. – DOI 10.1007/3–540–33213–8_4. – ISBN 978–3–540–33212–1, S. 91–122

[Goelzer u. a. 1995] GOELZER, Berenice ; HANSEN, Colin H. ; SEHRNDT, Gustav A.: *Occupational Exposure to Noise: Evaluation, Prevention and Control*. World Health Organization, 1995 http://www.who.int/occupational_health/publications/occupnoise/en/#

[Gutin 2008] GUTIN, Gregory: *Digraphs Theory, Algorithms and Applications*. Springer, 2008. – ISBN 978–1–84800–997–4

[Hahn u. Tretter 1973] HAHN, W ; TRETTER, S: Optimum processing for delay-vector estimation in passive signal arrays. In: *IEEE Transactions on Information Theory* 19 (1973), Nr. 5, S. 608–614. http://dx.doi.org/10.1109/TIT.1973.1055077. – DOI 10.1109/TIT.1973.1055077. – ISSN 0018–9448

[Heilemann 2012] HEILEMANN, Steffen: *Untersuchung unterschiedlicher Konfigurationsparameter bei der TDOA-basierten Sprecherlokalisierung*, Universität Stuttgart, Bachelorarbeit, 2012

[Hopcroft u. Tarjan 1973] HOPCROFT, John ; TARJAN, Robert: Algorithm 447: Efficient Algorithms for Graph Manipulation. In: *Communications of the ACM* 16 (1973), Nr. 6, S. 372–378. http://dx.doi.org/10.1145/362248.362272. – DOI 10.1145/362248.362272. – ISSN 0001–0782

[Hu u. Yang 2010] HU, Jwu-Sheng ; YANG, Chia-Hsin: Estimation of Sound Source Number and Directions under a Multisource Reverberant Environment. In: *EURASIP Journal on Advances in Signal Processing* (2010), Nr. 1, S. 1–14. http://dx.doi.org/10.1155/2010/870756. – DOI 10.1155/2010/870756. – ISSN 1687–6180

[Hu u. a. 2009] HU, Jwu-Sheng ; YANG, Chia-Hsing ; WANG, Cheng-Kang: Estimation of sound source number and directions under a multi-source environment. In: *International Conference on Intelligent Robots and Systems* (2009), S. 181–186. http://dx.doi.org/10.1109/IROS.2009.5354706. – DOI 10.1109/IROS.2009.5354706. ISBN 978–1–4244–3803–7

[Huang u. Chi 2012] HUANG, Kuan-Lang ; CHI, Tai-Shih: TDOA information based vad for robust speech recognition in directional and diffuse noise field. In: *In-*

ternational Symposium on Chinese Spoken Language Processing (2012), S. 126–130. http://dx.doi.org/10.1109/ISCSLP.2012.6423514. – DOI 10.1109/ISCSLP.2012.6423514

[Huang u. a. 2000] HUANG, Yiteng ; BENESTY, Jacob ; ELKO, Gary W.: Passive acoustic source localization for video camera steering. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* (2000), S. 909–912. http://dx.doi.org/10.1109/ICASSP.2000.859108. – DOI 10.1109/ICASSP.2000.859108. – ISBN 0780362934

[Huang u. a. 2004] HUANG, Yiteng ; BENESTY, Jacob ; ELKO, Gary W.: *Audio Signal Processing For Next-Generation Multimedia Communication Systems*. Springer, 2004. – ISBN 978–1–4020–7768–5

[Huang u. a. 2001] HUANG, Yiteng ; BENESTY, Jacob ; ELKO, Gary W. ; MERSEREAU, Russell M.: Real-Time Passive Source Localization : A Practical Linear-correction Least-squares Approach. In: *IEEE Transactions on Speech and Audio Processing* 9 (2001), Nr. 8, S. 943–956. http://dx.doi.org/10.1109/89.966097. – DOI 10.1109/89.966097. – ISSN 1063–6676

[Jacob 2000] JACOB, Benesty: Adaptive eigenvalue decomposition algorithm for passive acoustic source localization. In: *The Journal of the Acoustical Society of America* 107 (2000), Nr. 1, S. 384–91. http://dx.doi.org/10.1121/1.428310. – DOI 10.1121/1.428310. – ISSN 1520–8524

[Ji u. Zha 2003] JI, X. ; ZHA, Hongyuan: Robust sensor localization algorithm in wireless ad-hoc sensor networks. In: *International Conference on Computer Communications and Networks* (2003), S. 527–532. http://dx.doi.org/10.1109/ICCCN.2003.1284219. – DOI 10.1109/ICCCN.2003.1284219. – ISSN 1095–2055

[Johnson 1975] JOHNSON, Donald B.: Finding all the elementary circuits of a directed graph. In: *SIAM Journal on Computing* 4 (1975), Nr. 1, S. 77–84. http://dx.doi.org/10.1137/0204007. – DOI 10.1137/0204007

[Jungnickel 2008] JUNGNICKEL, Dieter: *Graphs, networks and algorithms*. Springer, 2008. – ISBN 978–3–540–72779–8

[Kaschub 2007] KASCHUB, Matthias: *Echtzeitsystem zur Sprecherlokalisierung*, Universität Stuttgart, Diplomarbeit, 2007

[Kirchhoff 1845] KIRCHHOFF, Gustav R.: Ueber den Durchgang eines elektrischen Stromes durch eine Ebene, insbesondere durch eine kreisförmige. In: *Annalen der Physik und Chemie* (1845), S. 497–514

[Knapp u. Carter 1976] KNAPP, Charles H. ; CARTER, G. C.: The Generalized Correlation Method for Estimation of Time Delay. In: *IEEE Transactions on Acoustics,*

Speech, and Signal Processing 2 (1976), S. 320–327. http://dx.doi.org/10.1109/TASSP.1976.1162830. – DOI 10.1109/TASSP.1976.1162830. – ISSN 0096–3518

[Köhler 2005] KÖHLER, B U.: *Konzepte der statistischen Signalverarbeitung*. Springer Verlag, 2005. – ISBN 3540234918

[Komplexitätszoo 2014] KOMPLEXITÄTSZOO: *www.complexityzoo.com*. Stand: Juli 2014

[Kreißig u. Yang 2010] KREISSIG, Martin ; YANG, Bin: A Graph Theoretical Framework for Consistent Time Differences of Arrival. In: *ITG-Fachbericht-Sprachkommunikation* (2010), S. 38

[Kreißig u. Yang 2011] KREISSIG, Martin ; YANG, Bin: Discussion on the bottom-up synthesis of consistent graphs: datemm revisited. In: *Signal Processing and Applied Mathematics for Electronics and Communications* (2011)

[Kreißig u. Yang 2012] KREISSIG, Martin ; YANG, Bin: An Efficient Algorithm for the Synthesis of Fully Consistent Graphs. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* (2012), S. 2653 – 2656. http://dx.doi.org/10.1109/ICASSP.2012.6288462. – DOI 10.1109/ICASSP.2012.6288462. – ISSN 1520–6149

[Kreißig u. Yang 2013] KREISSIG, Martin ; YANG, Bin: Fast and Reliable TDOA Assignment in Multi-Source Reverberant Environments. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* (2013), S. 355–359. http://dx.doi.org/10.1109/ICASSP.2013.6637668. – DOI 10.1109/ICASSP.2013.6637668. – ISSN 1520–6149

[Kruskal 1956] KRUSKAL, Joseph B.: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. In: *Proceedings of the American Mathematical Society* 7 (1956), Nr. 1, S. 48–50. http://dx.doi.org/10.2307/2033241. – DOI 10.2307/2033241. – ISSN 00029939

[Lee u. Choi 2010] LEE, Byoung-gi ; CHOI, JongSuk: Multi-source sound localization using the competitive k-means clustering. In: *IEEE Conference on Emerging Technologies & Factory Automation* 2 (2010), Nr. 1, S. 1–7. http://dx.doi.org/10.1109/ETFA.2010.5641169. – DOI 10.1109/ETFA.2010.5641169. ISBN 978–1–4244–6848–5

[Lehmann u. Johansson 2008] LEHMANN, E. ; JOHANSSON, A.: Prediction of energy decay in room impulse responses simulated with an image-source model. In: *Journal of the Acoustical Society of America* 124 (2008), S. 269–277. http://dx.doi.org/10.1121/1.2936367. – DOI 10.1121/1.2936367

[Lehmann 2014] LEHMANN, Eric: *http://www.eric-lehmann.com/CodeFiles/ISM.zip*. Stand: Juli 2014

[Liu u. a. 2012] LIU, Chunshan ; ZAKHAROV, Y.V. ; CHEN, Teyan: Broadband Underwater Localization of Multiple Sources Using Basis Pursuit De-Noising. In: *IEEE Transactions on Signal Processing* 60 (2012), Nr. 4, S. 1708–1717. http://dx.doi.org/10.1109/TSP.2011.2181506. – DOI 10.1109/TSP.2011.2181506. – ISSN 1053–587X

[Lloyd 1982] LLOYD, S.: Least squares quantization in PCM. In: *IEEE Transactions on Information Theory* 28 (1982), Nr. 2, S. 129–137. http://dx.doi.org/10.1109/TIT.1982.1056489. – DOI 10.1109/TIT.1982.1056489. – ISSN 0018–9448

[Loesch 2013] LOESCH, Benedikt: *Complex Blind Source Separation with Audio Applications*, Universität Stuttgart, Doktorarbeit, 2013

[Loesch u. a. 2010] LOESCH, Benedikt ; EBRAHIM, Parisa ; YANG, Bin: Comparison of different algorithms for acoustic source localization. In: *ITG-Fachbericht-Sprachkommunikation* (2010)

[Loesch u. a. 2009] LOESCH, Benedikt ; UHLICH, Stefan ; YANG, Bin: Multidimensional Localization of Multiple Sound Sources using Frequency Domain ICA and an Extended State Coherence Transform. In: *IEEE Workshop on Statistical Signal Processing* (2009), S. 677–680. http://dx.doi.org/10.1109/SSP.2009.5278486. – DOI 10.1109/SSP.2009.5278486. ISBN 978–1–4244–2709–3

[Nesta u. Omologo 2012a] NESTA, Francesco ; OMOLOGO, Maurizio: Enhanced multidimensional spatial functions for unambiguous localization of multiple sparse acoustic sources. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* (2012), Nr. 1, S. 213–216. http://dx.doi.org/10.1109/ICASSP.2012.6287855. – DOI 10.1109/ICASSP.2012.6287855. ISBN 978–1–4673–0046–9

[Nesta u. Omologo 2012b] NESTA, Francesco ; OMOLOGO, Maurizio: Generalized State Coherence Transform for Multidimensional TDOA Estimation of Multiple Sources. In: *IEEE Transactions on Audio, Speech, and Language Processing* 20 (2012), Nr. 1, S. 246–260. http://dx.doi.org/10.1109/TASL.2011.2160168. – DOI 10.1109/TASL.2011.2160168. – ISSN 1558–7916

[Oppenheim u. Schafer 1999] OPPENHEIM, Alan V. ; SCHAFER, Ronald W.: *Discrete-Time Signal Processing*. 2. Prentice-Hall, 1999. – ISBN 978–0131988422

[Oualil u. a. 2013] OUALIL, Youssef ; MAGIMAI.-DOSS, Mathew ; FAUBEL, Friedrich ; KLAKOW, Dietrich: A prababilistic framework for multiple speaker localization. In: *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing* (2013), S. 3962–3966. http://dx.doi.org/10.1109/ICASSP.2013.6638402. – DOI 10.1109/ICASSP.2013.6638402. – ISSN 1520–6149

[Oxley 2011] OXLEY, James: *Matroid Theory*. 2. Oxford University Press, 2011. – ISBN 978–0–19–960339–8

[Pferschy u. Schauer 2009] PFERSCHY, Ulrich ; SCHAUER, Joachim: The Knapsack Problem with Conflict Graphs. In: *Journal of Graph Algorithms and Applications* 13 (2009), Nr. 2, S. 233–249. http://dx.doi.org/10.7155/jgaa.00186. – DOI 10.7155/jgaa.00186. – ISSN 1526–1719

[Quazi 1981] QUAZI, Azizul H.: An Overview on the Time Delay Estimate in Active and Passive Systems for Target Localization. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* (1981), S. 527–533. http://dx.doi.org/10.1109/TASSP.1981.1163618. – DOI 10.1109/TASSP.1981.1163618. – ISSN 0096–3518

[Rebennack 2009] REBENNACK, Steffen: Stable Set Problem: Branch and Cut Algorithms. Version: 2009. http://dx.doi.org/10.1007/978-0-387-74759-0_634. In: FLOUDAS, Christodoulos A. (Hrsg.) ; PARDALOS, Panos M. (Hrsg.): *Encyclopedia of Optimization*. Springer Link, 2009. – DOI 10.1007/978–0–387–74759–0_634. – ISBN 978–0–387–74758–3, S. 3676–3688

[Roth 1971] ROTH, P. R.: Effective measurements using digital signal analysis. In: *IEEE Spectrum* (1971), S. 62–70. http://dx.doi.org/10.1109/MSPEC.1971.5218046. – DOI 10.1109/MSPEC.1971.5218046. – ISSN 0018–9235

[Russell u. Norvig 2010] RUSSELL, Stuart ; NORVIG, Peter: *Artificial Intelligence: A modern approach*. Pearson Education, 2010. – ISBN 0137903952

[Scheuing 2007] SCHEUING, Jan: *Lokalisierung von Sprechern durch mehrkanalige Verarbeitung akustischer Signale*, Universität Stuttgart, Doktorarbeit, 2007

[Scheuing u. Yang 2008] SCHEUING, Jan ; YANG, Bin: Disambiguation of TDOA Estimation for Multiple Sources in Reverberant Environments. In: *IEEE Transactions on Audio, Speech, and Language Processing* 16 (2008), Nr. 8, S. 1479–1489. http://dx.doi.org/10.1109/TASL.2008.2004533. – DOI 10.1109/TASL.2008.2004533. – ISSN 1558–7916

[Schmidt 1986] SCHMIDT, Ralph O.: Multiple Emitter Location and Signal Parameter Estimation. In: *IEEE Transactions on Antennas and Propagation* 34 (1986), Nr. 3, S. 276–280. http://dx.doi.org/10.1109/TAP.1986.1143830. – DOI 10.1109/TAP.1986.1143830. – ISSN 0018–926X

[Schmidt 1996] SCHMIDT, Ralph O.: Least Squares Range Difference Location. In: *IEEE Transactions on Aerospace and Electronic Systems* 32 (1996), Nr. 1, S. 234–242. http://dx.doi.org/10.1109/7.481265. – DOI 10.1109/7.481265. – ISSN 0018–9251

[Schroeder 1965] SCHROEDER, M R.: New Method of Measuring Reverberation Time. In: *Journal of the Acoustical Society of America* 37 (1965), Nr. 6, S. 1187–1188. http://dx.doi.org/10.1121/1.1909343. – DOI 10.1121/1.1909343

[Shen u. a. 2008] SHEN, Guowei ; ZETIK, Rudolf ; THOMÄ, Reiner S.: Performance Comparison of TOA and TDOA Based Location Estimation Algorithms in LOS Environment. In: *Workshop on positioning, navigation and communication* 2008 (2008), S. 71–78. ISBN 9781424417995

[Sidorovich u. Gershman 1998] SIDOROVICH, D. V. ; GERSHMAN, A. B.: Two-dimensional wideband interpolated root- MUSIC applied to measured seismic data. In: *IEEE Transactions on Signal Processing* 46 (1998), Nr. 8, S. 2263–2267. http://dx.doi.org/10.1109/78.705460. – DOI 10.1109/78.705460. – ISSN 1053–587X

[Smith u. Abel 1987] SMITH, J. ; ABEL, J.: Closed-form least-squares source location estimation from range-difference measurements. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 35 (1987), Nr. 12, S. 1661–1669. http://dx.doi.org/10.1109/TASSP.1987.1165089. – DOI 10.1109/TASSP.1987.1165089. – ISSN 0096–3518

[So u. a. 2008] SO, Hing C. ; CHAN, Yiu T. ; CHAN, Frankie Kit W.: Closed-Form Formulae for Time-Difference-of-Arrival Estimation. In: *IEEE Transactions on Signal Processing* 56 (2008), Nr. 6, S. 2614–2620. http://dx.doi.org/10.1109/TSP.2007.914342. – DOI 10.1109/TSP.2007.914342. – ISSN 1053–587X

[Stoica u. Li 2006] STOICA, Petre ; LI, Jian: Lecture Notes - Source Localization from Range-Difference Measurements. In: *IEEE Signal Processing Magazine* 23 (2006), Nr. 6, S. 63–66. http://dx.doi.org/10.1109/SP-M.2006.248717. – DOI 10.1109/SP–M.2006.248717. – ISSN 10535888

[Tack 2009] TACK, Guido: *Constraint Propagation - Models, Techniques, Implementation*, Universität des Saarlandes, Dissertation, 2009

[Tarjan 1972] TARJAN, Robert E.: Depth-First Search and Linear Graph Algorithms. In: *SIAM Journal on Computing* 1 (1972), Nr. 2, S. 146–160. http://dx.doi.org/10.1137/0201010. – DOI 10.1137/0201010. – ISSN 0097–5397

[Tarjan u. Read 1975] TARJAN, Robert E. ; READ, R C.: Bounds on Backtrack Algorithms for listing Cycles, Paths, and Spanning Trees. In: *Networks* (1975)

[Tarjan u. Trojanowski 1976] TARJAN, Robert E. ; TROJANOWSKI, Anthony E.: Finding a maximum independent set. In: *Science* (1976), Nr. June

[Tutte 2001] TUTTE, William T.: *Graph Theory*. Cambridge University Press, 2001. – ISBN 9780521794893

[Vary u. a. 1998] VARY, Peter ; HEUTE, Ulrich ; HESS, Wolfgang: *Digitale Sprachsignalverarbeitung*. Teubner, 1998. – ISBN 9783519061656

[Vorländer u. Mechel 2002] VORLÄNDER, M. ; MECHEL, F.: *Room acoustics*. Springer Verlag, 2002. – 774–845 S. – ISBN 3540768327

[Yang u. Kreißig 2011] YANG, Bin ; KREISSIG, Martin: An Introduction to Consistent Graphs and Their Signal Processing Applications. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* (2011), S. 2740–2743. http://dx.doi.org/10.1109/ICASSP.2011.5947052. – DOI 10.1109/ICASSP.2011.5947052. ISBN 9781457705397

[Yang u. Kreißig 2013] YANG, Bin ; KREISSIG, Martin: A Graph-Based Approach to Assist TDOA Based Localization. In: *International Workshop on Multidimensional Systems* (2013), S. 1–6. ISBN 978–3–8007–3543–3

[Yang u. Scheuing 2005] YANG, Bin ; SCHEUING, Jan: Cramer-Rao Bound and Optimum Sensor Array For Source Localization From Time Differences of Arrival. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* 4 (2005), S. 961–964. http://dx.doi.org/10.1109/ICASSP.2005.1416170. – DOI 10.1109/ICASSP.2005.1416170. ISBN 0–7803–8874–7

[Yang u. Scheuing 2006] YANG, Bin ; SCHEUING, Jan: A theoretical analysis of 2D sensor arrays for TDOA based localization. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* 4 (2006), Nr. 5, S. 901–904. http://dx.doi.org/10.1109/ICASSP.2006.1661115. – DOI 10.1109/ICASSP.2006.1661115