

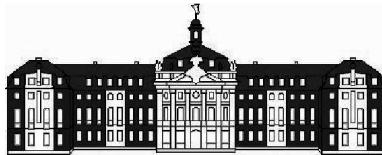
# Matching von Baumstrukturen in der medizinischen Bildverarbeitung

von  
Jan Hendrik Metzen

Diplomarbeit  
im Studiengang Diplom-Informatik

angefertigt am  
Institut für Informatik  
Westfälische Wilhelms-Universität Münster

mit Unterstützung von  
Center for Medical Diagnostic Systems and Visualization,  
MeVis GmbH, an der Universität Bremen



## **Erstprüfer**

Prof. Dr. Xiaoyi Jiang  
WWU Münster

## **Zweitprüfer**

Prof. Dr. Heinz-Otto Peitgen  
MeVis

## **Betreuer**

Dr. Tim Kröger (CeVis)  
Andrea Schenk (MeVis)  
Dr. Stephan Zidowitz (MeVis)

Münster, den 10. Juli 2006



## Zusammenfassung

In vielen medizinischen Anwendungen ist eine Registrierung verschiedener Bilddatensätze desselben Organs sinnvoll. Häufig geschieht eine solche Registrierung mit Hilfe manuell in den Bilddatensätze platzierter Landmarken. In dieser Arbeit wird ein Verfahren vorgestellt, mit dem automatisiert sinnvolle Landmarken bestimmt werden können. Dazu werden Knoten der zuvor mittels eines Segmentierungsverfahrens extrahierten Gefäßbäume mit Hilfe des so genannten Assoziationsgraph-Verfahrens einander zugeordnet und die Koordinaten der so zugeordneten Knoten als Landmarken benutzt. Abschließend wird skizziert, wie aus diesen Landmarken und den Intensitäten der Bilddaten eine elastische Registrierung gewonnen werden kann.



## Danksagung

Ich danke Prof. Dr. Xiaoyi Jiang für das Stellen des sehr interessanten Themas für diese Diplomarbeit. Die Aufmerksamkeit, Sorgfalt und Ausdauer, mit der er mich betreut hat, war eine große Unterstützung bei der Erstellung dieser Arbeit.

Meinen Betreuern bei MeVis, Andrea Schenk, Dr. Stephan Zidowitz und Dr. Tim Kröger, danke ich für die zahlreichen konstruktiven Ratschläge, die wertvollen Ideen und die anregenden Besprechungen. Ihre Anmerkungen waren mir eine große Hilfe bei der Erstellung dieser Ausarbeitung.

Meinen Eltern und meiner Freundin Silke danke ich für ihre Unterstützung während meines Studiums und für das Korrekturlesen dieser Arbeit.



# Inhaltsverzeichnis

<b>Zusammenfassung</b>	<b>iii</b>
<b>Danksagung</b>	<b>v</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Aufgabenstellung . . . . .	1
1.2. Inhalt dieser Arbeit . . . . .	2
1.3. Gewinnung der Baumstrukturen . . . . .	2
1.4. Zielsetzung . . . . .	3
1.5. Gliederung . . . . .	4
<b>2. Verfahren zum Matchen von Bäumen</b>	<b>5</b>
2.1. Strukturbasierte Ansätze . . . . .	6
2.1.1. Editierdistanz von Bäumen . . . . .	6
2.1.2. Modellbasiertes Matching von Bäumen . . . . .	7
2.2. Lokale Ansätze . . . . .	9
2.3. Globale Ansätze . . . . .	10
2.3.1. Gewichtetes bipartites Matching . . . . .	11
2.3.2. Assoziationsgraph . . . . .	12
2.4. Vergleich der Verfahren . . . . .	18
<b>3. Matching von Baumstrukturen aus Leber und Lunge</b>	<b>19</b>
3.1. Vorverarbeitung . . . . .	19
3.2. Constraints zur Erzeugung des Assoziationsgraphen . . . . .	23
3.2.1. Unäre Constraints . . . . .	24
3.2.2. Binäre Constraints . . . . .	33
3.2.3. Bewertungsfunktion für Constraints . . . . .	36
3.2.4. Wahl eines Referenzknotens . . . . .	36
3.3. Augmentierung der Clique . . . . .	37
<b>4. Bestimmung einer maximalen Clique</b>	<b>41</b>
4.1. Komplexitätstheoretische Überlegungen . . . . .	41
4.2. Kontinuierliche Charakterisierung des <i>Maximum Clique Problems</i> . . . . .	42
4.3. Heuristische Verfahren . . . . .	42
4.3.1. Greedy Algorithmus . . . . .	43
4.3.2. Evolutionäre Spieltheorie . . . . .	43
4.3.3. Genetische Algorithmen . . . . .	46

4.4. Vergleich der verschiedenen Verfahren . . . . .	47
<b>5. Anwendung auf Gefäßsysteme in Leber und Lunge</b>	<b>49</b>
5.1. Visualisierungen . . . . .	49
5.1.1. Darstellung des Matchings . . . . .	49
5.1.2. Übertragung einer Baumfärbung . . . . .	50
5.2. Evaluierung . . . . .	51
5.2.1. Ground Truth . . . . .	52
5.2.2. Optimierung der Constraints mittels Ground-Truth Daten . .	52
5.2.3. Analyse verschiedener Konfigurationsmöglichkeiten . . . . .	55
5.3. Ergebnisse . . . . .	57
5.4. Bewertung . . . . .	58
<b>6. Registrierung</b>	<b>61</b>
6.1. Landmarkenbasierte Registrierung . . . . .	62
6.2. Intensitätsbasierte Registrierung . . . . .	62
6.3. Ein kombiniertes Verfahren . . . . .	63
6.4. Beurteilung . . . . .	65
<b>7. Ausblick</b>	<b>67</b>
<b>A. Graphentheorie</b>	<b>69</b>
<b>Eidesstattliche Erklärung</b>	<b>ix</b>
<b>Abbildungsverzeichnis</b>	<b>xiii</b>
<b>Tabellenverzeichnis</b>	<b>xv</b>
<b>Index</b>	<b>xvii</b>
<b>Literaturverzeichnis</b>	<b>xix</b>



# Kapitel 1.

## Einleitung

### 1.1. Aufgabenstellung

Sowohl von der Leber als auch von der Lunge können mit bildgebenden Verfahren wie Computertomographie (CT) oder Magnetresonanztomographie (MRT) dreidimensionale, digitale Abbilder erstellt werden. In vielen medizinischen Anwendungen ist es wünschenswert, dem Anwender mehrere Aufnahmen desselben Organs zur Verfügung zu stellen, um ihm möglichst viele relevante Informationen zugänglich zu machen. Dies kann beispielsweise in den folgenden Situationen sinnvoll sein:

- Eine Lunge soll sowohl im eingeatmeten Zustand als auch im ausgeatmeten Zustand untersucht werden.
- Ein Organ wird sowohl mit CT als auch mit MRT aufgenommen, um die Vorteile beider Verfahren auszunutzen.
- Es werden Verlaufsuntersuchungen eines Organs (beispielsweise bei einem Tumorbefund) über mehrere Wochen, mit regelmäßigen Aufnahmen des Organs, vorgenommen.
- Die Regeneration der Leber nach einer Leberlebendspende (Living Donor Liver Transplantation, LDLT) wird verfolgt.

Aufgrund von Einflüssen wie Atmung und Herzschlag und verschiedener Positionen des Patienten während der Aufnahmen kann es zwischen zwei Aufnahmen jedoch zu erheblichen Lageveränderungen und Deformationen der jeweiligen Organe kommen. Dadurch ist es häufig schwierig, Stellen, die denselben Bereich eines Organs in den Bildern repräsentieren, zu identifizieren. Dies gilt insbesondere in Situationen, in denen sich die Anatomie des zugrunde liegenden Organs selbst erheblich ändert, wie zum Beispiel nach einer Leberresektion<sup>1</sup> oder der Regeneration der Leber nach einer Leberlebendspende. Eine weitere Schwierigkeit können Datensätze, die mittels unterschiedlicher bildgebender Verfahren wie CT und MRT gewonnen wurden, darstellen, da diese von deutlich abweichender Qualität sein können.

---

<sup>1</sup>Unter einer Leberresektion versteht man die operative Entfernung kranker Gewebeteile der Leber, zum Beispiel nach einem Tumorbefall.

Eine solche Zuordnung verschiedener Darstellungen eines Organs bezeichnet man als *Registrierung*. Nach [HBHH01] berechnet eine Registrierung die räumliche Ausrichtung zweier Bilder, die nötig ist, um diese Bilder miteinander vergleichen und in dasselbe Koordinatensystem transformieren zu können.

Um eine Registrierung der Datensätze automatisiert zu ermöglichen, gilt es, nach Eigenschaften der Organe zu suchen, die unter den oben genannten Bedingungen vergleichsweise invariant sind. Ein geeigneter Ansatzpunkt scheinen hierfür die verschiedenen Gefäßsysteme innerhalb eines Organs zu sein. Diese verändern zwar ebenfalls Lage und Ausdehnung, jedoch bleibt ihre Struktur nahezu unverändert. Auch bei anatomischen Veränderungen wie der Regeneration der Leber verbleiben relativ große Teile der Gefäßsysteme unverändert. Gelingt es, korrespondierende Teilbereiche dieser Strukturen zu identifizieren, so kann mittels dieser strukturellen Informationen die Registrierung der Bilddaten vereinfacht werden.

## 1.2. Inhalt dieser Arbeit

Das Thema der vorliegenden Arbeit ist das *Matchen* (Zuordnen) gewisser Strukturen innerhalb von Leber oder Lunge. Bei diesen Strukturen handelt es sich um Gefäßsysteme wie den Bronchialbaum der Lunge oder Pfortader, Leberarterie bzw. hepatische Vene der Leber. Diese Strukturen wurden zuvor mittels eines mehrschrittigen Verfahrens – vergleiche Abschnitt 1.3 – gewonnen und sind in einer abstrakten Graph-Repräsentation gegeben. Die Matching-Verfahren arbeiten allein auf diesem Graphen; die ursprünglichen Bilddaten werden an keiner Stelle benutzt.

Da es sich bei den zugrunde liegenden Strukturen um Bäume handelt, kommen verschiedene Verfahren zum Matchen von Bäumen als Ansatzpunkt in Betracht. Diese werden in dieser Arbeit vorgestellt und ihre jeweiligen Vor- und Nachteile diskutiert. Im Weiteren wurde eines dieser Verfahren in C++ implementiert und in *MeVisLab*, einer Softwareplattform zur medizinischen Bildverarbeitung und Visualisierung, integriert. Die Ergebnisse dieses Matching-Verfahrens werden vorgestellt und bewertet. Als Ausblick wird skizziert, wie aus dem Matching der Gefäßbäume eine Registrierung der Bilddaten abgeleitet werden kann.

## 1.3. Gewinnung der Baumstrukturen

Von großer Bedeutung für die Qualität des Matchings der Gefäßbäume und somit auch der Registrierung der Bilddaten ist die Güte der detektierten Gefäßsysteme. Die Gewinnung der Gefäßbäume ist bereits in *MeVisLab* realisiert. Sie verläuft in mehreren Schritten:

In der Aufnahme des Organs wird mittels eines Schwellwertverfahrens zunächst das Gefäßsystem segmentiert. Des Weiteren wird auf diesem Gefäßsystem eine Mittelachsentransformation durchgeführt. Daraufhin wird aus den bestimmten Mittelachsen eine Graphstruktur abgeleitet. Die Knoten dieser Graphstrukturen entspre-

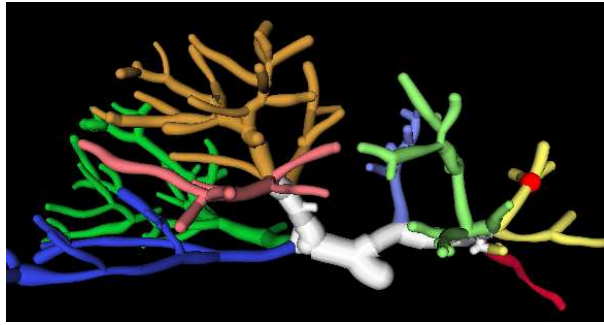


Abbildung 1.1.: Baumstruktur einer Portalvene

Die Wurzel befindet sich unten mittig (weiß), eingefärbt ist der Baum gemäß einer Einteilung der Leber in Segmente gemäß Couinaud/Bismuth.

chen dabei genau den Stellen, an denen sich eine Mittelachse verzweigt, die Kanten genau einer Mittelachse zwischen zwei solchen Verzweigungsstellen. Für jede Kante werden bestimmte Größen wie Länge und Volumen des korrespondierenden Gefäßabschnitts berechnet und als Attribute der Kanten abgelegt. Das Ergebnis ist ein gewichteter Graph. In diesem Graphen werden schließlich die baumartigen Strukturen detektiert, die als Grundlage für den Matching-Prozess dienen können. Für eine detailliertere Erläuterung der Detektion von Gefäßbäumen in *MeVisLab* sei auf [Sel00] verwiesen.

Das Ergebnis eines solchen Baumextraktionsverfahrens ist in Abbildung 1.1 dargestellt. In diesem Fall handelt es sich um eine Portalvene. Deutlich zu erkennen sind die größeren Durchmesser der Kanten nahe der Wurzel (dem weiß-gefärbten Bereich des Baums). Die Einfärbung des Baumes entspricht einer Einteilung der Leber in Segmente gemäß Couinaud/Bismuth [Cou57].

## 1.4. Zielsetzung

Das zu implementierende Matching-Verfahren soll mit geringem Zeitaufwand für vergleichbare Bäume einen relevanten Anteil der Knoten sinnvoll zuordnen.

- Unter *geringem Zeitaufwand* ist dabei eine Laufzeit des Verfahrens von wenigen Minuten für Bäume mit bis zu 500 Knoten auf einem handelsüblichen PC zu verstehen.
- Unter *vergleichbaren Bäumen* verstehen wir Bäume, die aus Aufnahmen desselben Organs ermittelt wurden. Dies schließt Aufnahmen wie die Verlaufsaufnahmen einer Leberregeneration mit ein, obwohl sich hier die zugrunde liegenden anatomischen Strukturen erheblich ändern können. Als nicht vergleichbar werden Aufnahmen derselben anatomischen Struktur aus Organen verschiedener Patienten angesehen. Ebenfalls nicht geplant ist eine Zuordnung

zwischen einem segmentierten Gefäßbaum und einem abstrakten Modell eben dieses Gefäßbaumes.

- Unter einem *relevanten Anteil der Knoten* ist zu verstehen, dass eine Mehrheit der Knoten aus Gefäßbereichen, die im anderen Baum eine Entsprechung haben, zugeordnet wird.
- Unter einer *sinnvollen Zuordnung* ist eine Zuordnung solcher Knoten zu verstehen, die demselben Verzweigungspunkt des zugrunde liegenden anatomischen Gefäßsystems entsprechen. Diese anatomische Korrektheit ist für die oben definierten vergleichbaren Bäumen eindeutig bestimmt.

## 1.5. Gliederung

In Kapitel 2 werden verschiedene Verfahren zum Matchen von Bäumen vorgestellt und ihre Vor- und Nachteile diskutiert. Eines dieser Verfahren – welches das Matching über einen so genannten Assoziationsgraphen gewinnt – wird in Kapitel 3 an den gegebenen Kontext des Matchings von Gefäßsystemen in Leber und Lunge angepasst. Dazu ist die Detektion einer maximalen Clique in einem Graphen notwendig. In Kapitel 4 werden verschiedene Lösungsverfahren für dieses klassische, graphentheoretische Problem vorgestellt und diskutiert. Die Ergebnisse des Matching-Verfahrens werden in Kapitel 5 vorgestellt und Stärken und Schwächen des Ansatzes für die gegebene Aufgabe beleuchtet. In Kapitel 6 wird schließlich erläutert, wie aus dem Matching der Gefäßbäume eine Registrierung der Bilddaten abgeleitet werden kann. In Appendix A werden grundlegende Begriffe aus der Graphentheorie kurz rekapituliert, die für diese Arbeit relevant sind.

## Kapitel 2.

# Verfahren zum Matchen von Bäumen

In diesem Kapitel sollen verschiedene Verfahren, die das *Matchen von Bäumen* adressieren, vorgestellt und die jeweiligen Vor- und Nachteile diskutiert werden. Unter einem Matching verstehen wir eine Menge von Zuordnungen von Knoten des einen Baums zu Knoten des anderen Baums. Hierbei wird jedem Knoten maximal ein Knoten des anderen Baums zugeordnet.

In gerichteten Bäumen (was gleichbedeutend ist mit Bäumen, in denen eine Wurzel ausgezeichnet ist) wird dadurch unmittelbar eine Zuordnung von Kanten der beiden Bäumen induziert: Zwei Kanten  $e_1 = (v_{11}, v_{12})$  und  $e_2 = (v_{21}, v_{22})$  werden einander genau dann zugeordnet, wenn ihre Endknoten  $v_{12}$  und  $v_{22}$  zugeordnet sind. Da in einem gerichteten Baum jeder Knoten maximal eine eingehende Kante besitzt, ist diese Zuordnung eindeutig.

Matchings von Bäumen, in denen einem Knoten des einen Baums ein Teilbaum des anderen zugeordnet wird (ein Teilbaum also gewissermaßen zu einem Knoten kontrahiert wird), werden nicht behandelt. In [PSZ01] wird auf diese Art von Many-to-Many Matchings eingegangen.

Im Rahmen dieses Kapitels wird unter einem Baum  $T = (V, E, w, c_v)$  stets ein zusammenhängender, azyklischer Graph mit Knotenmenge  $V$ , Kantenmenge  $E$ , einem als Wurzel ausgezeichneten Knoten  $w$  und einer Funktion  $c_v : V \rightarrow \mathbb{R}^n$  verstanden. Bei  $c_v$  handelt es sich um eine Funktion, die jedem Knoten  $v \in V$  eine Menge von  $n$  Attributen zuordnet. Die Kanten sind mithin unattributiert; dies ist jedoch keine Einschränkung, da sich Kantenattribute gegebenenfalls in Knotenattribute umwandeln lassen, indem man die Attribute einer Kante zu den Attributen ihres Endknotens hinzufügt.

Die verschiedenen Verfahren lassen sich in drei Kategorien einteilen. Zum einen sind *lokale Ansätze* zu nennen. Bei diesen wird in einem iterativen Verfahren versucht, nach und nach Knoten der beiden Bäume einander zuzuordnen. In der Regel wird dabei auf beiden Bäumen eine Art Tiefendurchlauf durchgeführt. Demgegenüber stehen *globale Ansätze*: In diesen werden implizit verschiedene Mengen von Zuordnungen generiert und bewertet und eine optimale Menge von Zuordnungen ausgewählt. Üblicherweise geschieht dies, indem aus den beiden Bäumen eine andere Datenstruktur abgeleitet wird, auf der eine zum ursprünglichen Matching-Problem

äquivalente Aufgabe gegeben ist. Aus der Lösung dieser Aufgabe lässt sich wiederum ein Matching ablesen. Als dritter Ansatz verbleiben verschiedene strukturbasierte Ansätze zum Matching. Unter diese fallen zum einen Verfahren zum Berechnen der *Editierdistanz* von Bäumen. In diesen wird der eine Baum durch eine Folge von Editieroperationen – wie zum Beispiel Einfügen und Löschen von Kanten – in den anderen umgeformt. Aus dieser Folge von Editieroperationen lässt sich schließlich eine Zuordnung von Knoten ableiten. Zum anderen gibt es Ansätze, parallel zu dem Matching die beiden Bäume auf einen gemeinsamen Grundbaum zurückzuführen. Basis dieser Verfahren ist ein *Deformationsmodell*, das beschreibt, welche Deformationen die Bäume von dem anatomisch zugrunde liegenden Baum verfremdet haben können.

## 2.1. Strukturbasierte Ansätze

### 2.1.1. Editierdistanz von Bäumen

Analog zur *Editierdistanz* von Strings (auch als Levenshtein-Distanz bekannt, vergleiche [Lev66]), lässt sich auch für Graphen – und somit insbesondere für Bäume – eine Editierdistanz definieren. Zur Bestimmung einer solchen Editierdistanz ist die Festlegung einer bestimmten Menge von *Editieroperationen* und von Kosten für jede solche Editieroperation notwendig. Beispiele für Editieroperationen sind das Einfügen/Löschen von Knoten/Kanten und das Ändern von Knoten- bzw. Kantenattributen. Für zwei gegebene Bäume  $T_1$  und  $T_2$  sei eine Folge  $S = [s_1, \dots, s_n]$  von Editieroperationen  $s_i$  gegeben, die  $T_1$  in  $T_2$  überführt. Da jede Editieroperation  $s_i$  Kosten  $c(s_i) \geq 0$  besitzt, gibt es unter allen solchen Editierfolgen beliebiger Länge eine mit minimalen Gesamtkosten  $c = \sum_i c(s_i)$ . Dieser Wert  $c$  wird als die Editierdistanz der beiden Bäume  $T_1$  und  $T_2$  bezeichnet. Die Editierdistanz hängt dabei natürlich von der Wahl der erlaubten Editieroperationen und den Kosten ab. In Abbildung 2.1 ist ein Beispiel für eine Editierfolge angegeben, die einen Baum  $T_1$  in einen Baum  $T_2$  überführt.

Ist eine optimale Editierfolge bekannt, so lässt sich aus dieser mit etwas zusätzlichem Aufwand eine Zuordnung von Baumknoten ableiten[Bil03]. Die wesentliche Schwierigkeit besteht nun darin, unter allen möglichen Editierfolgen diejenige mit minimalen Kosten zu finden. Während sich die Editierdistanz für zwei String der Länge  $n$  und  $m$  in  $O(nm)$  Zeit berechnen lässt, ist die Berechnung der Editierdistanz von zwei allgemeinen Graphen als  $\mathcal{NP}$ -vollständig bekannt. Ob sich die Editierdistanz zweier Bäume in polynomieller Zeit berechnen lässt, hängt im Wesentlichen davon ab, ob man die Bäume als *geordnet* ansehen kann. Ein Baum heißt geordnet, wenn unter den Kindern jedes seiner Knoten eine Ordnung ausgezeichnet ist, wenn es also zum Beispiel sinnvoll ist, von einem linken und einem rechten Kindknoten zu sprechen. Im Falle von Gefäßbäumen von Leber und Lunge ist die Definition einer Ordnung der Kindknoten jedoch aufgrund der Beweglichkeit der Gefäßsysteme und eventueller Segmentierungsfehler schwierig. Die Berechnung der Editierdistanz von

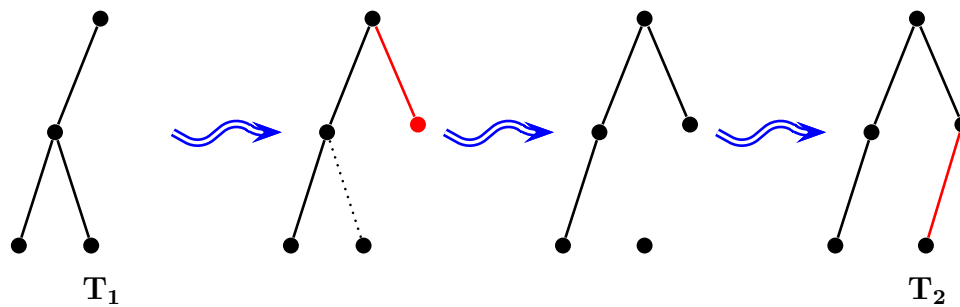


Abbildung 2.1.: Beispiel einer Editierfolge

1. Einfügen eines neuen Knotens und einer verbindenden Kante (rot).
2. Entfernen der gestrichelten Kante.
3. Einfügen einer neuen verbindenden Kante (rot)

ungeordneten Bäumen ist wiederum  $\mathcal{NP}$ -vollständig; allerdings existieren gewisse beschränkte Editierdistanzen<sup>1</sup>, deren Berechnung in polynomieller Zeit möglich ist. Die Beschränkungen erscheinen aber für die gegebene Anwendung als zu restriktiv. Für eine ausführliche Behandlung des Baum-Editierdistanz-Problems sei auf [Bil03] verwiesen.

### 2.1.2. Modellbasiertes Matching von Bäumen

Die verschiedenen Ansätze zur Berechnung einer Editierdistanz erscheinen für die konkrete Anwendung entweder als zu restriktiv oder als zu ineffizient (siehe oben). Abhilfe kann ein Verfahren liefern, das explizit die Ursachen für die Abweichungen der Baumstrukturen berücksichtigt. Ein solches wird in [GH06] vorgeschlagen. Der Ansatz soll im folgenden skizziert werden; für eine ausführlichere und formale Darstellung sei auf obiges Paper verwiesen.

Dem *Modellbasierten Matching von anatomischen Bäumen* liegt die Überlegung zugrunde, dass die beiden zu matchenden Bäume – wie in Abschnitt 1.4 gefordert – auf derselben anatomischen Gefäßstruktur basieren. Topologische Unterschiede der segmentierten Bäume gegenüber diesem anatomischen Baum resultieren somit fast ausschließlich aus Problemen des Segmentierungsverfahren oder der Mittelachsen-transformation aufgrund schlechter Bilddaten. In [GH06] werden zwei Deformationsmodelle angegeben, die typische Verfälschungen der Topologie, die während dieser Schritte auftreten können, formal modellieren. Jedes dieser Deformationsmodelle lässt nur eine Untermenge der möglichen Matchings zu. Diese Deformationsmodelle werden am Ende dieses Abschnitts vorgestellt.

Weiterhin wird eine Menge von Ähnlichkeitsmaßen, basierend auf geometrischen Attributen wie Kantenlängen, Verzweigungswinkeln und relativen Koordinaten von

<sup>1</sup>Bei einer beschränkten Editierdistanz werden Editierfolgen, die gewisse Eigenschaften nicht erfüllen, von vornherein nicht untersucht. Daher ist die resultierende beschränkte Editierdistanz in der Regel größer als die allgemeine Editierdistanz.

Verzweigungspunkten, vorgestellt. Auf eine genauere Vorstellung wird hier verzichtet, die Ähnlichkeitsmaße ähneln jedoch stark den Bewertungsfunktionen der Constraints aus Abschnitt 3.2. Jedem, der mit einem Deformationsmodell verträglichen Matchings, wird somit eine Bewertung bezüglich dieser Ähnlichkeitsmaße zugeordnet. Mit Hilfe eines "dynamic programming" Algorithmus, der in [GH06] vorgestellt wird, kann das global optimale Matching zweier Bäume (bezüglich dieses Ähnlichkeitsmaßes) mit  $n$  bzw.  $m$  Knoten in  $O(n^2m^2)$  Zeit berechnet werden.

Dem *primären Deformationsmodell* liegt folgende Überlegung zugrunde: Durch Probleme während der Extraktion der Bäume aus den Bilddaten kann es dazu kommen, dass erstens Teiläste, die im anatomischen Baum existieren, im extrahierten Baum fehlen und zweitens, umgekehrt zusätzliche Äste im extrahierten Baum auftauchen, die keine reale anatomische Entsprechung besitzen. Beide Arten von Fehlern können dazu führen, dass in einem der zu matchenden Bäume Äste vorhanden sind, die im anderen nicht vorkommen. Das primäre Deformationsmodell erlaubt nun genau solche Matchings von Baumknoten, die eine Reduktion der beiden Bäume  $T_1$  und  $T_2$  auf einen sogenannten *Grundbaum* induzieren. Dieser Grundbaum  $T_C$  muss die Eigenschaft haben, dass man sowohl  $T_1$  als auch  $T_2$  nur durch Hinzufügen zusätzlicher Äste und der zugehörigen Bifurkationsknoten von  $T_C$  ableiten kann. Eine formale Definition, welche Matchings einen solchen Grundbaum induzieren, wird in [GH06] gegeben, das Prinzip lässt sich jedoch auch aus Abbildung 2.2 erkennen. Jedem Paar von Zuordnungen entspricht hierbei ein Knoten des Grundbaums und die Kanten des Grundbaums ergeben sich unmittelbar aus der Struktur der Bäume  $T_1$  und  $T_2$ .  $T_1$  und  $T_2$  lassen sich offenbar durch Hinzufügen von Kanten und Bifurkationsknoten aus  $T_C$  ableiten.

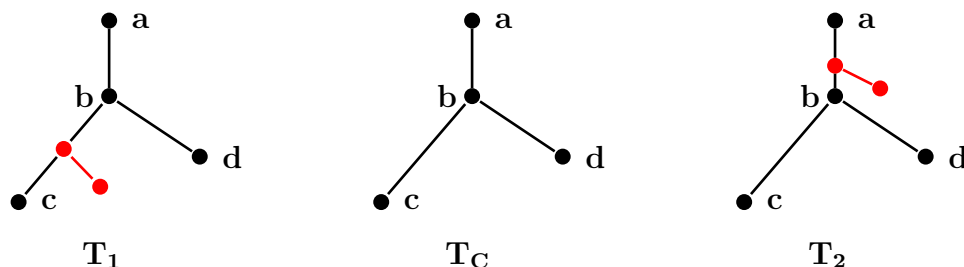


Abbildung 2.2.: Ein Grundbaum des primären Deformationsmodells

Zwei zu matchende Bäume  $T_1$  und  $T_2$ , sowie der Grundbaum  $T_C$ .

Die rot gefärbten Kanten und Knoten sind im Grundbaum nicht enthalten und besitzen keine Entsprechung im jeweils anderen Baum. Zugeordnet sind jeweils die gleich bezeichneten Knoten, analoges gilt für die Knoten des Grundbaums.

Das *sekundäre Deformationsmodell* berücksichtigt zusätzlich zu den Fehlern, die das primäre Deformationsmodell adressiert, noch das folgende Problem: Die Detektion von Bifurkationsknoten ist mitunter nicht stabil. So können zwei dicht benach-



barte Bifurkationsknoten des anatomischen Baumes als ein Trifurkationsknoten<sup>2</sup> erkannt werden oder umgekehrt ein Trifurkationsknoten des anatomischen Baumes als zwei Bifurkationen. Ebenso könnte die topologische Ordnung von zwei Bifurkationen vertauscht zu werden. Das sekundäre Deformationsmodell berücksichtigt diese Fehlerquelle bei der Ermittlung des Grundbaums. Ein Grundbaum des sekundären Deformationsmodells ist in Abbildung 2.3 dargestellt.

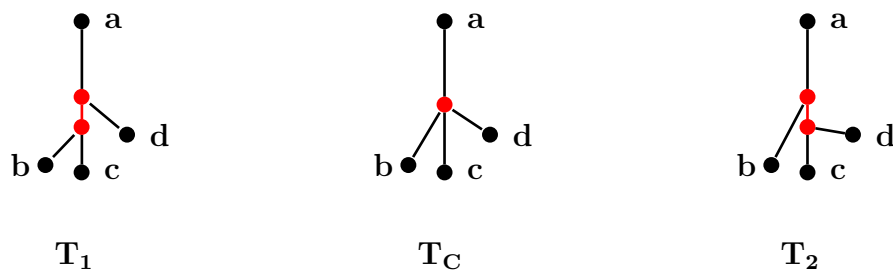


Abbildung 2.3.: Ein Grundbaum des sekundären Deformationsmodells

Zwei zu matchende Bäume  $T_1$  und  $T_2$ , sowie der Grundbaum  $T_C$ .

Die rot gefärbten Kanten und Knoten sind im Grundbaum in einem Knoten zusammengefasst. Zugeordnet seien hierbei jeweils die gleich bezeichneten Knoten, analoges gilt für die Knoten des Grundbaums.

## 2.2. Lokale Ansätze

Im Folgenden wird exemplarisch ein lokales Verfahren zum Matchen von Bäumen vorgestellt. Es orientiert sich an dem in [CAM<sup>+</sup>05] erläuterten Verfahren, von welchem hier nur die grundlegenden Ideen veranschaulicht werden sollen. Im Gegensatz zu den anderen Verfahren werden hier nicht nur Knoten einander zugeordnet, sondern auch Kanten.

Das Verfahren führt auf beiden Bäumen eine parallele Tiefensuche durch. Es wird davon ausgegangen, dass die beiden Wurzeln der Bäume richtig erkannt wurden, daher werden die Wurzelknoten einander zugeordnet. Im Weiteren werden folgende zwei Schritte wiederholt durchgeführt, um eine Menge möglichst guter Matchinghypothesen zu generieren. Diese können schließlich bezüglich der globalen Qualität ihrer Knoten- und Kantenzuordnungen bewertet werden und die Beste als Matching ausgewählt werden.

1. Seien  $v_1$  und  $v_2$  die Knoten, die als letztes einander zugeordnet wurden. Seien  $e_{11}, \dots, e_{1n}$  die ausgehenden Kanten von  $v_1$  und  $e_{21}, \dots, e_{2m}$  die von  $v_2$ . Generiere alle möglichen Zuordnungsmengen von Kanten aus  $e_{11}, \dots, e_{1n}$  mit Kanten aus

<sup>2</sup>Ein Trifurkationsknoten ist ein Knoten mit drei Kindknoten.

$e_{21}, \dots, e_{2m}$  und wähle die  $k_1$  besten aus (Die Kriterien für Optimalität werden später definiert). Führe für jede dieser  $k_1$  Hypothesen den 2. Schritt durch.

2. Für jedes Paar von Kanten  $e_i$  und  $e_j$  aus der jeweiligen Hypothese aus Schritt 1: Seien  $T(e_i)$  und  $T(e_j)$  die Teilbäume, die an den Endknoten von  $e_i$  und  $e_j$  wurzeln. Wähle die  $k_2$  besten Zuordnungen von einem Knoten aus dem Teilbaum  $T(e_i)$  zu einem Knoten aus  $T(e_j)$  aus. Führe für jede dieser  $k_2$  Zuordnungshypothesen rekursiv Schritt 1 aus.

Dieses Verfahren terminiert, da Blattknoten keine ausgehenden Kanten haben und die Rekursion daher nach endlich vielen Durchgängen jeweils in Schritt 1 terminiert. Es wird auf diesem Wege eine endliche Menge von Matchinghypothesen generiert.

Falls  $k_1 = k_2 = 1$  so wird nur eine Matchinghypothesen generiert, indem beim Tiefendurchlauf jeweils die lokal beste Auswahl getroffen wird. Dies ist sehr schnell, jedoch auch sehr empfindlich gegenüber Segmentierungsartefakten, die zu Fehlzuordnungen im oberen Bereich des Baumes führen können. Daher ist es sinnvoll,  $k_1$  und  $k_2$  größer als 1 zu wählen. Die lokal beste Zuordnung mag zwar aufgrund von Segmentierungsartefakten und Rauschen fehlerhaft sein, unter den  $k_1$  bzw.  $k_2$  lokal besten wird jedoch in der Regel zumindest eine sein, die sich auch im weiteren Verfahren als sinnvoll erweist.

Werden  $k_1$  und  $k_2$  hingegen zu groß gewählt, so werden praktisch alle möglichen Zuordnungsmengen von Knoten und Kanten generiert. Dies ist jedoch aufgrund der enormen Anzahl möglicher Matchings (nämlich exponentiell vieler) nicht praktikabel. Bei geeigneter Wahl von  $k_1$  und  $k_2$  und sinnvollen Entscheidungskriterien für die Auswahl von Knoten- bzw. Kantenzuordnungen in Schritt 1 und 2 wird eine hinreichend kleine Menge guter Matchinghypothesen generiert.

Es verbleibt, Kriterien aufzustellen, wie Knoten- und Kantenzuordnungen bewertet werden können. Für eine formale Definition solcher Kriterien sei auf [CAM<sup>+</sup>05] verwiesen. Für den Fall der Leber- und Lungengefäße ist es sinnvoll, vor allem Kanten zuzuordnen die ähnliche Form und Radius besitzen, sowie Knotenzuordnungsmengen zu begünstigen, die möglichst viele Knoten der beiden Bäume matchen. Dies korrespondiert bei den lokalen Entscheidungen (Schritt 2) dazu, Knoten möglichst weit oben in den Bäumen  $T(e_i)$  und  $T(e_j)$  zu wählen.

## 2.3. Globale Ansätze

Im Folgenden werden zwei Beispiele von globalen Matching-Verfahren vorgestellt. Bei beiden Verfahren werden implizit verschiedene Mengen von Zuordnungen generiert und bewertet und eine optimale Menge von Zuordnungen ausgewählt. Dies geschieht, indem aus den beiden Bäumen eine andere Graphstruktur abgeleitet wird, auf der eine zum ursprünglichen Matching-Problem äquivalente Aufgabe gegeben ist, aus deren Lösung sich ein Matching ablesen lässt.

### 2.3.1. Gewichtetes bipartites Matching

In dem in diesem Abschnitt vorgestellten Verfahren wird das Problem des Matchens zunächst auf ein anderes Problem transformiert. Dazu wird aus den beiden Bäumen  $T_1 = (V_1, E_1, w_1, c_{v_1})$  und  $T_2 = (V_2, E_2, w_2, c_{v_2})$  ein weiterer Graph  $G_B$  konstruiert.

$G_B = (V_B, E_B, c_B)$  wird wie folgt konstruiert: Die Knotenmenge  $V_B$  sei die Vereinigung der Knotenmengen  $V_1$  und  $V_2$  der beiden Bäume:  $V_B = V_1 \cup V_2$ . Jeder Knoten aus  $V_1$  sei mit jedem Knoten aus  $V_2$  verbunden, also  $E_B = V_1 \times V_2$ . Die Kantengewichtungsfunktion  $c_B$  ordnet jeder Kante  $e = (v_1, v_2)$  eine Güte zu, die von den Knotenattributen  $c_{v_1}(v_1)$  und  $c_{v_2}(v_2)$  abhängt. Anschaulich gilt: Je größer  $c_B(e)$ , desto passender ist die Zuordnung  $v_1 \leftrightarrow v_2$ . Bei  $G_B$  handelt es sich also um einen *gewichteten, bipartiten Graphen*. In Abbildung 2.4 sind zwei Bäume und der konstruierte bipartite Graph (jeweils ohne Gewichte) dargestellt.

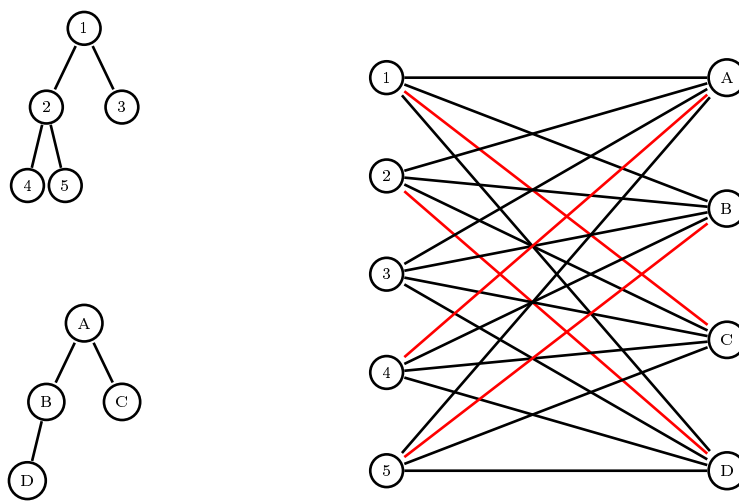


Abbildung 2.4.: Zwei Bäume und der konstruierte bipartite Graph

Links sind die beiden zu matchenden Bäume dargestellt, rechts der aus diesen konstruierte bipartite Graph. Rot eingefärbt ist in diesem ein maximales bipartites Matching.

Unter einem *bipartiten Matching* versteht man die Auswahl einer Teilmenge  $E_M \subseteq E_B$ , sodass gilt:

$$\forall e_1 = (v_{11}, v_{12}), e_2 = (v_{21}, v_{22}) \in E_M \text{ gilt: } e_1 \neq e_2 \Rightarrow v_{11} \neq v_{21} \wedge v_{12} \neq v_{22} \quad (2.1)$$

Ein bipartites Matching heißt *maximal*, wenn es unter allen solchen Matchings maximale Kardinalität  $|E_M|$  besitzt. Offenbar ist die Kardinalität eines solchen Matchings gerade  $\min(|V_1|, |V_2|)$ . Die rot gefärbten Kanten in Abbildung 2.4 sind ein solches

maximales, bipartites Matching. Aus dem bipartiten Matching kann man unmittelbar ein Matching der zugrunde liegenden Bäume ableiten: Falls  $e = (v_1, v_2) \in E_M$  so werden die Knoten  $v_1$  und  $v_2$  aus den beiden Bäumen einander zugeordnet.

Fordert man statt maximaler Kardinalität  $|E_M|$ , dass  $\sum_{e \in E_M} c_B(e)$  maximal wird, so spricht man von einem *gewichteten bipartiten Matching*. Da  $c_B(e)$  ja gerade die Güte des Matchings  $v_1 \leftrightarrow v_2$  war, ist das resultierende Matching der Bäume das optimale bezüglich der Gütefunktion  $c_B$ .

Verfahren zur Bestimmung maximaler, gewichteter bipartiter Matchings sind hinlänglich bekannt, als Quelle sei Lawler [Law76] genannt. Die Laufzeit-Komplexität ist  $O(n^3)$ . Nachteilig beim bipartiten Matching ist jedoch, dass die Topologie der Bäume nicht explizit repräsentiert werden kann. Angenommen im Beispiel 2.5 habe die Kante  $e = (4, D)$  in  $G_B$  hohes Gewicht, sodass die Zuordnung  $4 \leftrightarrow D$  gewählt wird. Aufgrund der Topologie der Bäume erscheint es nahe liegend dann auch die Zuordnung  $5 \leftrightarrow E$  zu wählen. Hat jedoch die Kante  $e_1 = (5, E)$  geringeres Gewicht als die Kante  $e_2 = (5, F)$ , so wird stattdessen die Zuordnung  $5 \leftrightarrow F$  getroffen.

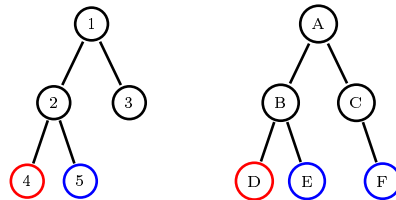


Abbildung 2.5.: Mögliche Matchings von Bäumen

Die beiden rot eingefärbten Knoten seien einander zugeordnet. Beim bipartiten Matching fällt es schwer zu entscheiden, welcher der beiden blau eingefärbten Knoten im rechten Baum dem im linken Baum zugeordnet werden sollte.

Dieses Problem ließe sich nur lösen, wenn nach jeder getroffenen Zuordnung zweier Knoten die Gewichte aller Kanten in  $G_B$  neu berechnet würden. Dies würde jedoch das Verfahren erheblich verkomplizieren. Stattdessen wird im nächsten Abschnitt das Matching mithilfe eines so genannten *Assoziationsgraphen* vorgestellt. Dieser folgt ähnlichen Ansätzen wie das bipartite Matching, jedoch ist auch die explizite Modellierung topologischer Eigenschaften möglich.

### 2.3.2. Assoziationsgraph

Im folgenden wird nun das oben erwähnte *Assoziationsgraph*-Verfahren vorgestellt. Ebenso wie beim bipartiten Matching wird zunächst ein anderer Graph konstruiert, der gewisse Beziehungen zwischen den Bäumen explizit modelliert. Bestimmte Strukturen innerhalb dieses Assoziationsgraphen korrespondieren dann zu einem

Matching der beiden ursprünglichen Bäume. Die Vorgehensweise richtet sich grob nach [Pel02]. Das Verfahren wird allerdings durch die zusätzliche Einführung expliziter Constraints<sup>3</sup> deutlich flexibler.

Die Knoten  $v_A \in V_A$  des Assoziationsgraphen  $G_A = (V_A, E_A)$  sind Paare von Knoten der beiden zugrunde liegenden Bäume  $T_1 = (V_1, E_1)$  und  $T_2 = (V_2, E_2)$ , d.h.  $V_A \subseteq V_1 \times V_2$ . Zur Definition des Assoziationsgraphen sei eine Menge *unärer Constraints*  $C_F = \{f_i\}$  mit  $f_i(v_A) \in \{true, false\}$  für  $v_A \in V_A$  und eine Menge *binärer Constraints*  $C_G = \{g_j\}$  mit  $g_j(v_A, v_B) \in \{true, false\}$  für  $v_A, v_B \in V_A$  gegeben. Die  $f_i$  und  $g_j$  hängen von topologischen Eigenschaften der Bäume sowie Knoten- und Kantenattributen ab. Die Wahl der  $f_i$  und  $g_j$  kann von Fall zu Fall variieren; wir gehen vorerst davon aus, dass geeignete Constraints gewählt wurden. Zur Wahl dieser Constraints sei auf Abschnitt 2.3.2 und Kapitel 3 verwiesen. Es folgt eine erste Version der Definition des Assoziationsgraphen.

**Definition 1** (Baum - Assoziationsgraph). *Seien  $T_1 = (V_1, E_1)$  und  $T_2 = (V_2, E_2)$  zwei Bäume. Dann definieren wir den Assoziationsgraph  $G_A = (V_A, E_A)$  von  $T_1$  und  $T_2$  wie folgt:*

1.  $V_A = \{v_A \in V_1 \times V_2 \mid \forall f_i \in C_F : f_i(v_A) = true\}$
2.  $E_A = \{e = (v_A, v_B) \in V_A \times V_A \mid \forall g_j \in C_G : g_j(v_A, v_B) = true\}$

Bei  $T_1$  und  $T_2$  wird es sich im Rahmen dieser Arbeit immer um Bäume handeln, jedoch ist Assoziationsgraph-Verfahren auch auf nicht-baumförmige Graphen anwendbar [BB76]. Ein Knoten  $v_A = (v_{A1}, v_{A2}) \in V_A$  des Assoziationsgraphen repräsentiert die Zuordnung  $v_{A1} \leftrightarrow v_{A2}$  zweier Knoten der Bäume. Die unären Constraints geben Bedingungen an, wann eine solche Zuordnung sinnvoll ist. Einem Knoten  $v_A$  kommt also eine ähnlich Bedeutung zu wie einer Kante des bipartiten Graphen im vorherigen Abschnitt.

Die im bipartiten Matching schwierige Berücksichtigung der Topologie der Bäume ergibt sich im Assoziationsgraphen auf natürliche Weise durch die Kanten  $E_A$ . Aus der Topologie der Bäume lassen sich Bedingungen ableiten, wann zwei Zuordnungen  $v_{A1} \leftrightarrow v_{A2}$  und  $v_{B1} \leftrightarrow v_{B2}$  *konsistent* sind. Eine solche Bedingung lässt sich in Form eines binären Constraints  $g_j$  kodieren, nämlich wie folgt: Für zwei Knoten  $v_A = (v_{A1}, v_{A2})$  und  $v_B = (v_{B1}, v_{B2})$  des Assoziationsgraphen sei  $g_j(v_A, v_B)$  genau dann *true*, wenn die Zuordnungen  $v_{A1} \leftrightarrow v_{A2}$  und  $v_{B1} \leftrightarrow v_{B2}$  konsistent sind.

Das Ziel des Matchings von Bäumen war, möglichst vielen Knoten  $v_1$  aus  $T_1$  "ähnliche" Knoten  $v_2$  aus  $T_2$  zu zuordnen, wobei die einzelnen Zuordnungen untereinander konsistent sein sollen. Die Ähnlichkeit von Knoten wird auf Ebene des Assoziationsgraphen durch die Menge der Knoten repräsentiert, die Verträglichkeit durch die Kanten zwischen zwei Knoten. Somit ist folgende Aufgabe äquivalent zum Suchen eines maximalen Matchings von Bäumen: Suche im Assoziationsgraphen  $G_A$  eine Teilmenge  $Q \subseteq V_A$  maximaler Kardinalität, für die gilt:

---

<sup>3</sup>Der Begriff Constraint wird im Folgenden genauer definiert. Constraints dienen dazu, gewisse Anforderungen an ein sinnvolles Matching zu formalisieren.

$$\forall v_A, v_B \in Q : v_A \neq v_B \Rightarrow (v_A, v_B) \in E \quad (2.2)$$

Eine solche Teilmenge ist genau eine *global maximale Clique* wie in Appendix A auf Seite 70 definiert. Das maximale Matching der Bäume lässt sich nun leicht aus der Clique ableiten: Für alle  $v_A = (v_{A1}, v_{A2}) \in Q$  nehme die Zuordnung  $v_{A1} \leftrightarrow v_{A2}$  zum Matching hinzu. Damit lässt sich das Assoziationsgraph-Verfahren wie in Prozedur AssocGraph zusammenfassen.

```

Input : Zwei Bäume  $T_1 = (V_1, E_1)$  und  $T_2 = (V_2, E_2)$ , Menge unärer
          Constraints  $C_F$ , Menge binärer Constraints  $C_g$ 
Output : Eine Matching  $M = \{v_1 \leftrightarrow v_2\}$  von  $T_1$  und  $T_2$ 
Setze  $V_A = \emptyset, E_A = \emptyset$ 
/* Berechne Knotenmenge  $V_A$  */
foreach  $v_i \in V_1$  do
    foreach  $v_j \in V_2$  do
        if  $\forall f_k \in C_F : f_k((v_i, v_j)) = \text{true}$  then
             $V_A \leftarrow V_A \cup \{(v_i, v_j)\}$ 
/* Berechne Kantenmenge  $E_A$  */
foreach  $v_i \in V_A$  do
    foreach  $v_j \in V_A$  do
        if  $v_i \neq v_j$  und  $\forall g_k \in C_G : g_k(v_i, v_j) = \text{true}$  then
             $E_A \leftarrow E_A \cup \{(v_i, v_j)\}$ 
/* Erzeuge Assoziationsgraph  $G_A$  */
 $G_A \leftarrow (V_A, E_A)$ 
/* Berechne maximale Clique  $Q$ , zum Beispiel mit Hilfe des
    Verfahrens GreedyClique in Kapitel 3 auf Seite 44 */
 $Q \leftarrow \text{GreedyClique}(G_A)$ 
/* Leite das Matching  $M$  aus der Clique  $Q$  ab */
 $M \leftarrow \emptyset$ 
foreach  $v = (v_1, v_2) \in Q$  do
     $M \leftarrow M \cup \{v_1 \leftrightarrow v_2\}$ 
return M;

```

#### Prozedur AssocGraph

### Ein Matching zweier unattributierter Bäume

Im Folgenden soll das Assoziationsgraph-Verfahren anhand zweier einfacher, nicht-attributierter Bäume veranschaulicht werden. Die beiden Bäume seien  $T_1$  und  $T_2$  aus Abbildung 2.6 auf der nächsten Seite.

In dieser – im Vergleich zur ursprünglichen Aufgabe des Matchings von Gefäßstrukturen – stark vereinfachten Situation können Constraints sich nur aus der To-

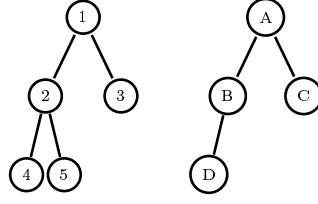


Abbildung 2.6.: Zwei unattributierte Bäume

pologie der Bäume ergeben. Es sei darauf hingewiesen, dass die hier vorgestellten Constraints im weiteren Verlauf der Arbeit nicht verwendet werden, da sie nur für idealisierte Bäume sinnvoll sind.

Wir wählen zunächst keine unären Constraints ( $C_F = \emptyset$ ) und zwei binäre Constraints ( $C_G = \{g_1, g_2\}$ ). Diese seien wie folgt definiert:

$$g_1((v_{A1}, v_{A2}), (v_{B1}, v_{B2})) = \text{true} \Leftrightarrow d_T(v_{A1}, \text{lca}(v_{A1}, v_{B1})) = d_T(v_{A2}, \text{lca}(v_{A2}, v_{B2})) \quad (2.3)$$

$$g_2((v_{A1}, v_{A2}), (v_{B1}, v_{B2})) = \text{true} \Leftrightarrow d_T(\text{lca}(v_{A1}, v_{B1}), v_{B1}) = d_T(\text{lca}(v_{A2}, v_{B2}), v_{B2}) \quad (2.4)$$

Hierbei bezeichne  $\text{lca}$  den "least common ancestor"<sup>4</sup> und  $d_t(v_1, v_2)$  die *topologische Distanz*<sup>5</sup> zweier Knoten im Baum, also die Anzahl Knoten auf dem eindeutigen Pfad von  $v_1$  nach  $v_2$ . Zwei Zuordnungen sind also konsistent, wenn die Pfade zwischen den beteiligten Knoten in beiden Bäumen dieselbe Länge besitzen<sup>6</sup>. Dies ist für die reale Anwendung viel zu restriktiv, wie in Abschnitt 3.2.2 erläutert wird.

Da  $C_F = \emptyset$ , gilt für die Knotenmenge  $V_A$  des Assoziationsgraphen:  $V_A = V_1 \times V_2$ . Die Kantenmenge  $E_A$  ist in Abbildung 2.7 auf der nächsten Seite zu sehen. Die Knoten  $D4 = (D, 4)$  und  $C3 = (C, 3)$  sind beispielsweise verbunden, da gilt:

$$\begin{aligned} d_T(D, \text{lca}(D, C)) &= d_T(D, A) = 2 = d_T(4, 1) = d_T(4, \text{lca}(4, 3)) \\ &\Rightarrow g_1((d, c), (4, 3)) = \text{true} \\ d_T(\text{lca}(D, C), C) &= d_T(A, C) = 1 = d_T(1, 3) = d_T(\text{lca}(4, 3), 3) \\ &\Rightarrow g_2((d, c), (4, 3)) = \text{true} \end{aligned}$$

Für die anderen Kanten gilt analoges. Eine Clique maximaler Kardinalität ist offenbar  $Q = \{A1, B2, C3, D4\}$ . Die diese Knoten verbindenden Kanten sind rot gefärbt. Aus dieser Clique lässt sich das Baum-Matching  $M = \{A \leftrightarrow 1, B \leftrightarrow 2, C \leftrightarrow$

<sup>4</sup>Als least common ancestor zweier Knoten  $v_1, v_2$  bezeichnet man den eindeutig bestimmten Knoten  $v$  auf dem Weg von  $v_1$  nach  $v_2$ , der minimales Level besitzt, vergleiche Appendix A auf Seite 71.

<sup>5</sup>Als topologische Distanz  $d_t$  zweier Knoten bezeichnet man die Anzahl der Knoten auf dem Pfad zwischen diesen, siehe Appendix A auf Seite 71.

<sup>6</sup>Genauer: Es müssen hierbei sowohl der im Baum aufwärts wie auch der abwärts führende Teil des Pfades in etwa gleich lang sein.

$3, D \leftrightarrow 4\}$  ableiten, die Knoten der Bäume sind dementsprechend gefärbt. Die global maximale Clique ist nicht eindeutig definiert, da die Clique  $Q' = \{A1, B2, C3, D5\}$  die gleiche Kardinalität wie  $Q$  besitzt. Aufgrund der gewählten Constraints sind die Zuordnungen  $D \leftrightarrow 4$  und  $D \leftrightarrow 5$  gleichwertig.

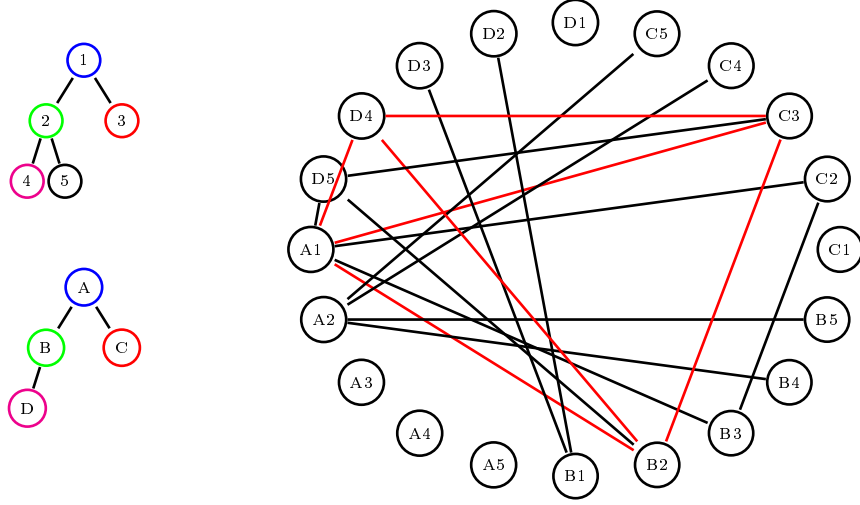


Abbildung 2.7.: Assoziationsgraph, eine maximale Clique und ein Matching  
Die maximale Clique ist rot gefärbt, das Matching durch übereinstimmende Farben in den beiden Bäumen ablesbar.

Betrachten wir nun die gleiche Situation, allerdings unter Berücksichtigung eines unären Constraints  $f_1$ , der wie folgt definiert ist:

$$f_1(v_A) = f_1((v_{A1}, v_{A2})) = \text{true} \Leftrightarrow |\text{level}(v_{A1}) - \text{level}(v_{A2})| \leq 1 \quad (2.5)$$

Das *Level*<sup>7</sup> der beiden Knoten darf also um maximal 1 differieren. Dem liegt die Überlegung zugrunde, dass zu matchende Knoten in den Bäumen in etwa auf demselben Level liegen sollten.

In Abbildung 2.8 auf der nächsten Seite ist der Assoziationsgraph bei Berücksichtigung des unären Constraints  $f_1$  zu sehen. Offenbar ändern sich die maximalen Cliques  $Q$  und  $Q'$  sowie die resultierenden Matchings nicht. Allerdings verringert sich die Größe des Assoziationsgraphen – bei größeren Bäumen würde diese Verringerung noch deutlich stärker ausfallen. Das ist wünschenswert, da das Bestimmen einer global maximalen Clique ein schwieriges Problem<sup>8</sup> ist. In Abschnitt 3.2.1 werden unäre Constraints vorgestellt, die nicht nur die Effizienz erhöhen, sondern auch die Qualität des Matchings verbessern.

<sup>7</sup>Als *Level* eines Knotens  $v$  bezeichnet man die Länge des (eindeutigen) Weges von der Wurzel  $w$  nach  $v$ , vergleiche Appendix A auf Seite 71.

<sup>8</sup>Es handelt sich um ein  $\mathcal{NP}$ -vollständiges Problem, wie in Abschnitt 4.1 auf Seite 41 erläutert wird.



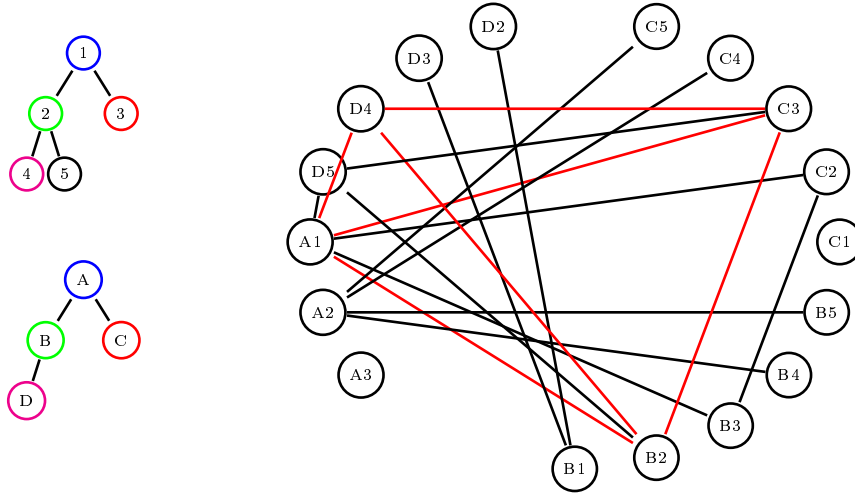


Abbildung 2.8.: Assoziationsgraph und maximale Clique bei Berücksichtigung von  $f_1$

### Modifikationen am Assoziationsgraphs-Verfahren

In der konkreten Anwendung – dem Matchen von Gefäßsystemen in Leber und Lunge – werden in Abschnitt 3.2 eine Reihe von unären und binären Constraints eingeführt. Nach den in Definition 1 auf Seite 13 eingeführten Bedingungen für  $V_A$  und  $E_A$  werden nur solche Knoten und Kanten aufgenommen, die sämtliche Constraints erfüllen. Häufig ist es aber wünschenswert auch Knoten (bzw. Kanten) aufzunehmen, die einem großen Teil der Constraints genügen, auch wenn sie eine Minderheit der Constraints nicht erfüllen.

Um eine höhere Flexibilität zu erreichen, passen wir die Constraints zunächst wie folgt an: Die unären Constraints seien eine Menge  $C_F = \{f_i\}$  mit  $f_i(v_A) \in [0, 1]$ , die binären Constraints eine Menge  $C_G = \{g_j\}$  mit  $g_j(v_A, v_B) \in [0, 1]$ . Dabei bedeutet  $f_i(v_A) = 1$  (bzw.  $g_j(v_A, v_B) = 1$ ), dass der jeweilige Constraint vollständig erfüllt ist.  $f_i(v_A) = 0$  (bzw.  $g_j(v_A, v_B) = 0$ ) bedeutet dagegen, dass der Constraint nicht erfüllt ist. Werte zwischen 0 und 1 entsprechen einer partielle Zustimmung eines Constraints. Für  $f_i(v_A) > 0.5$  (bzw.  $g_j(v_A, v_B) > 0.5$ ) überwiegt dabei die Zustimmung, bei Werten  $\leq 0.5$  die Ablehnung.

Weiterhin unterscheiden wir die Wichtigkeit der einzelnen Constraints; dazu erhält jeder Constraint eine bestimmte Anzahl *Stimmen*  $\omega_i \in \mathbb{N}_0$ ; je mehr Stimmen, umso höher wird der Constraint gewichtet. Welche Knoten (bzw. Kanten) in den Assoziationsgraphen aufgenommen werden, wird in einer *Abstimmung* bestimmt, bei der die durchschnittliche Bewertung aller Constraints für einen Knoten (bzw. eine Kante) mindestens 0.5 sein muss:

**Definition 2** (Baum - Assoziationsgraph (modifiziert)). Seien  $T_1 = (V_1, E_1)$  und  $T_2 = (V_2, E_2)$  zwei Bäume. Dann definieren wir den Assoziationsgraph  $G = (V_A, E_A)$  von  $T_1$  und  $T_2$  wie folgt:

$$1. V_A = \{v_A \in V_1 \times V_2 \mid \frac{\sum_{f_i \in C_F} \omega_i f_i(v_A)}{\sum_{f_i \in C_F} \omega_i} \geq 0.5\}$$

$$2. E_A = \{e = (v_A, v_B) \in V_A \times V_A \mid \frac{\sum_{g_j \in C_G} \omega_j g_j(v_A, v_B)}{\sum_{g_j \in C_G} \omega_j} \geq 0.5\}$$

Des Weiteren ist es möglich, die Knotenkardinalität  $|V_A|$  des Assoziationsgraphen zu verringern. Hierzu betrachte man die Mengen  $V_A[v_i] = \{v_A = (v_i, v_j) \in V_A \mid v_j \in V_2\}$ . Bei  $V_A[v_i]$  handelt es sich also um alle Knoten des Assoziationsgraphen, die eine potentielle Zuordnung von  $v_i \in V_1$  repräsentieren. Es gilt  $\bigcup_{v_i \in V_1} V_A[v_i] = V_A$ . Jedem

$v \in V_A[v_i]$  ist durch  $\frac{\sum_{f_i \in C_F} \omega_i f_i(v)}{\sum_{f_i \in C_F} \omega_i}$  eine Bewertung gegeben. Es bietet sich nun an, nur solche  $v_A = (v_i, v_j)$  in  $V_A[v_i]$  aufzunehmen, die unter allen  $v_j \in V_2$  eine möglichst gute Bewertung haben. Nimmt man maximal die  $|V_2|/10$  besten aller solcher  $v_A$  auf<sup>9</sup>, so verringert sich die Laufzeit des Verfahrens erheblich. Häufig erhöht sich gleichzeitig sogar die Qualität des Matchings, da die heuristischen Verfahren aus Kapitel 4 für kleine Graphen gegen bessere Lösungen konvergieren.

## 2.4. Vergleich der Verfahren

Da das exakte Berechnen einer – im Anwendungskontext sinnvollen – Editierdistanz zweier Bäume  $\mathcal{NP}$ -vollständig ist, verbliebe für diesen Ansatz nur, sich auf sehr kleine Bäume zu beschränken oder eine gute Heuristik zur Approximation der Editierdistanz zu finden. Selbst wenn ein solches Approximationsverfahren hinreichend gut wäre, wäre es schwierig, aus der approximierten Editierdistanz ein Matching der Baumknoten abzuleiten. Daher wurde dieser Ansatz nicht weiter verfolgt. Viel versprechender ist der modellbasierte Matchingansatz. Dieser wurde jedoch erst veröffentlicht, als diese Arbeit bereits in einem fortgeschrittenen Stadium war, sodass eine genauere Analyse nicht mehr möglich war.

Unter den vorgestellten globalen Ansätzen ist das Assoziationsgraph-Verfahren dem bipartiten Matching vorzuziehen, da sich in diesem topologische Eigenschaften für das Matching einfacher berücksichtigen lassen. Außerdem ist das Assoziationsgraph-Verfahren aufgrund der frei wählbaren Constraints sehr flexibel. So lassen sich durch die Wahl geeigneter Constraints auch Vorteile der lokalen Verfahren integrieren, wie in Abschnitt 3.2.1 beschrieben wird. Als Nachteil verbleibt beim Assoziationsgraph-Verfahren der Blackbox-Charakter, d. h. es ist in der Regel schwierig nachzuvollziehen, warum das Verfahren gewisse Knotenzuordnungen getroffen hat. Die Vorteile überwiegen jedoch, sodass im folgenden dieses Verfahren weiterverfolgt und eine Anpassung des Verfahrens an das Matching von Gefäßbäumen vorgestellt wird.

<sup>9</sup>Es werden natürlich auch weiterhin nur solche  $v_A$  aufgenommen, die auch gemäß Definition 2 in  $V$  sind.

## Kapitel 3.

# Matching von Baumstrukturen aus Leber und Lunge

Nachdem in Kapitel 2.3.2 das Assoziationsgraph-Verfahren auf einer allgemeinen Ebene eingeführt wurde, wird in diesem Kapitel eine speziell auf die Aufgabenstellung abgestimmte Variante vorgestellt. Dazu werden in Abschnitt 3.1 die Eingabedaten in eine einheitliche Form gebracht. Für diese Daten wird dann der Assoziationsgraph gemäß gewisser Constraints erzeugt. Die Qualität des Matchings hängt wesentlich von der Wahl der Constraints ab – eine Reihe von Constraints werden in Abschnitt 3.2 vorgestellt. Nach der Konstruktion des Assoziationsgraphen wird in diesem eine maximale Clique detektiert. Mögliche Ansätze hierfür werden in Kapitel 4 diskutiert. Das mittels des Assoziationsgraph-Verfahren gewonnene Matching lässt sich durch bestimmte Augmentierungsmethoden – die in Abschnitt 3.3 vorgestellt werden – noch verbessern.

### 3.1. Vorverarbeitung

Die meisten der in Kapitel 2 vorgestellten Verfahren benötigen als Eingabe einen *Baum*. Bei den zugrunde liegenden anatomischen Strukturen wie Leberpfortader und Bronchialsystem handelt es sich um solche baumförmigen Strukturen, sodass die Verfahren aus Kapitel 2 anwendbar sein sollten. Allerdings können während der Gewinnung der Daten verschiedene Probleme auftreten:

1. Um das gesamte Gefäßsystem zu erfassen, muss die Aufnahme genau zu dem Zeitpunkt gemacht werden, zu dem das Kontrastmittel sich im gesamten Gefäßsystem verteilt hat. Wird die Aufnahme zu früh gemacht, so kann es sein, dass das Kontrastmittel periphere Gefäßabschnitte noch nicht hinreichend durchdrungen hat. Wird die Aufnahme dagegen zu spät gemacht, so kann das Kontrastmittel aus den oberen Bereichen des Systems bereits wieder abgeflossen sein. Dies hat zur Folge, dass Wurzel und wurzelnahe Teile des Baums nicht repräsentiert sind, sodass der Baum in mehrere Zusammenhangskomponenten zerfällt.
2. Aufgrund der begrenzten Auflösung der bildgebenden Verfahren, Bewegungen des Patienten und Rauschen während der Aufnahme kann es dazu kommen,

dass das Segmentierungsverfahren in den Bilddaten vermeintliche Gefäßäste erkennt, die nicht real existieren. Weiterhin können Gefäßäste, die aufgrund ihrer Größe vom Segmentierungsverfahren gefunden werden sollten, unerkannt bleiben.

3. Aus ähnlichen Gründen kann es dazu kommen, dass der Verlauf der Gefäßbahnen lokal nicht eindeutig detektierbar ist, sodass während der Segmentierung eventuell eine anatomische Gefäßkante zweifach detektiert wird. Dies führt zu Zyklen, die allerdings lokal begrenzt sind.
4. In der Leber sind oft mehrere Gefäßsysteme gleichzeitig kontrastiert. Periphere Äste der Pfortader und Lebervene können sich dabei scheinbar berühren. Zwar werden Algorithmen zur Trennung dieser Gefäßsysteme durchgeführt, dennoch kann es in seltenen Fällen zu falschen Zuordnungen von Segmenten zu einem Gefäßsystem kommen.

Aufgrund dieser Probleme kann nicht davon ausgegangen werden, dass die Eingabe-Graphen stets baumförmig sind. Allerdings lässt sich mittels der folgenden Vorverarbeitungsschritte eine baumförmige Struktur gewinnen. Der Graph hat nach den Vorverarbeitungsschritten folgende Eigenschaften:

- Es gibt genau eine Zusammenhangskomponente und somit eine eindeutige Wurzel.
- Der Pfad eines jeden Knotens zur Wurzel ist eindeutig, d.h. es gibt keine Zyklen.

### Auflösen von Zyklen

Ein Baum mit Wurzel induziert automatisch eine Ausrichtung seiner Kanten (vergleiche Seite 71) dergestalt, dass jedem Knoten ein Level zugeordnet wird und jeder Knoten – mit Ausnahme der Wurzel – einen eindeutigen Vorgängerknoten besitzt, nämlich denjenigen seiner Nachbarknoten, der ein geringeres Level besitzt.

Ferner gilt: Ein gerichteter Graph, in dem jeder Knoten maximal einen Vorgänger hat und in dem von einem bestimmten Knoten  $w$  alle anderen Knoten  $v$  erreichbar sind, kann keine Zyklen enthalten und lässt sich somit als Baum mit Wurzel  $w$  interpretieren. Dies sieht man wie folgt: Sei ein solcher Graph gegeben. Da von  $w$  jeder andere Knoten erreichbar ist, indem man Kanten entlang ihrer Ausrichtung folgt, muss andererseits  $w$  von jedem Knoten  $v$  erreichbar sein, indem man Kanten entgegen ihrer Ausrichtung folgt. Somit gibt es eine Folge von Knoten  $v = v_1, v_2, \dots, v_n = w$  mit der Eigenschaft, dass  $v_{i+1}$  Vorgänger von  $v_i$  ist. Würde andererseits ein Zyklus existieren, der den Knoten  $v$  beinhaltet, so würde eine Folge von Knoten  $v = v_1^*, v_2^*, \dots, v_m^* = v$  existieren, für die ebenfalls gilt, dass  $v_{i+1}^*$  Vorgänger von  $v_i^*$  ist. Da  $w$  keinen Vorgänger besitzt muss gelten:  $\forall i \in \{1, \dots, m\} : w \neq v_i^*$ . Somit muss es einen Index  $j$  geben, für den gilt  $v_j \neq v_j^*$ . Sei  $j$  minimal mit dieser

Eigenschaft. Dann hat der Knoten  $v_{j-1}$  zwei Vorgänger, nämlich  $v_j$  und  $v_j^*$ . Widerspruch!

Bei den *Zusammenhangskomponenten* des Eingabe-Graphen handelt es sich genau um solche Strukturen; die zusammenhängenden Teilgraphen sind gerichtet und es existiert ein Knoten  $w$  mit obigen Eigenschaften. Sollen aus einer solchen Zusammenhangskomponente also alle Zyklen entfernt werden, so genügt es für jeden Knoten, alle bis auf eine Vorgängerkante zu löschen. Es verbleibt die Wahl der Kante, die nicht gelöscht werden soll. Hier ist es sinnvoll, eine Kante, deren zugeordnetes Volumen<sup>1</sup> möglichst groß ist, beizubehalten, da Kanten mit geringem Volumen wahrscheinlicher aus Segmentierungsfehlern oder Rauschen herrühren. Gleichzeitig muss allerdings auch garantiert sein, dass beim Löschen der überzähligen Kanten der Teilgraph zusammenhängend bleibt.

Dies kann man wie folgt erreichen: Von der Wurzel des Teilgraphs ausgehend traversiert man die jeweilige Zusammenhangskomponente in einem Breitendurchlauf (*Breadth-First-Order*). Bei jedem Knoten behält man die Kante aufrecht, die den Knoten mit dem bereits traversierten Teil des Teilgraph verbindet und unter allen solchen Kanten das maximale Volumen besitzt.

### Löschen überzähliger Zusammenhangskomponenten

Nach dem Entfernen der Zyklen handelt es sich bei jeder Zusammenhangskomponente des Graphen um einen Baum, bei dem Graphen also um einen *Wald*. Für die Verfahren aus Kapitel 2 wird als Eingabe jedoch ein einzelner Baum benötigt. Um dies zu erreichen gibt es zwei Möglichkeiten: Entweder man entfernt alle bis auf eine Zusammenhangskomponente oder man fügt einen zusätzlichen "Superknoten" ein, der als Vaterknoten der Wurzeln aller Zusammenhangskomponenten fungiert.

In der Regel sind aber alle bis auf eine Zusammenhangskomponente sehr klein<sup>2</sup>, sodass es akzeptabel ist, alle Zusammenhangskomponenten bis auf die größte zu löschen. Im Allgemeinen dürfte es sich bei sehr kleinen Zusammenhangskomponenten auch um Segmentierungsartefakte handeln.

### Pruning des Baums

Nach den obigen Schritten handelt es sich bei dem Graphen um einen Baum. In diesem Baum kann es jedoch noch Segmentierungsartefakte geben, die keine reale, anatomische Entsprechung haben. Ohne die zugrunde liegenden Bilddaten ist es prinzipiell nicht möglich, diese fälschlicherweise detektierten Gefäßabschnitte festzustellen und dann zu entfernen. Häufig ist die Ursache für diese Segmentierungsartefakte Rauschen, was die Vermutung nahe legt, dass diese Artefakte vergleichsweise klein sind. Daher wird zum *Pruning* folgende Heuristik benutzt:

---

<sup>1</sup>Als das einer Kante zugeordnete Volumen wird das Volumen des korrespondierenden Gefäßabschnitts in den Bilddaten verstanden.

<sup>2</sup>Die Größe einer Zusammenhangskomponente ist hierbei als die Summe der Längen aller ihrer Kanten definiert.

Für alle Knoten  $v$  des Baums:

Falls die Größe des von  $v$  induzierten Teilbaums zuzüglich der in  $v$  eingehenden Kante weniger als einen bestimmten Anteil (eine gute Wahl ist ungefähr 10 Prozent) der Größe des von  $\text{pred}(v)$ <sup>3</sup> induzierten Teilbaums besitzt, so entferne diesen von  $v$  induzierten Teilbaum und die eingehende Kante.

Es werden also Teilbäume entfernt, die im Vergleich zu ihren "Geschwisterbäumen" klein sind. Dieser Ansatz funktioniert gut, wenn die meisten Verzweigungspunkte der zugrunde liegenden Gefäßsysteme *dichotom* sind. Eine Verzweigung heißt dichotom, wenn sich das Gefäßsystem an dieser Stelle in zwei etwa gleich große Gefäßsysteme verzweigt. In Bronchialbäumen der Lunge ist diese Bedingung in der Regel erfüllt, sodass hier das Pruning zu sinnvollen Ergebnissen führt, wie in Abschnitt 5.2.3 gezeigt wird. Die Gefäßsysteme der Leber dagegen sind nicht dichotom; daher ist es hier meist ratsam auf das Pruning zu verzichten, da sonst auch Teilbäume entfernt würden, die reale, anatomische Entsprechungen besitzen.

In Abbildung 3.1 ist der gleiche Bronchialbaum einmal vor und einmal nach dem Pruning dargestellt. Deutlich zu Erkennen ist die Entfernung vieler kurzer, abzweigender Äste vom eingehenden Gefäß.

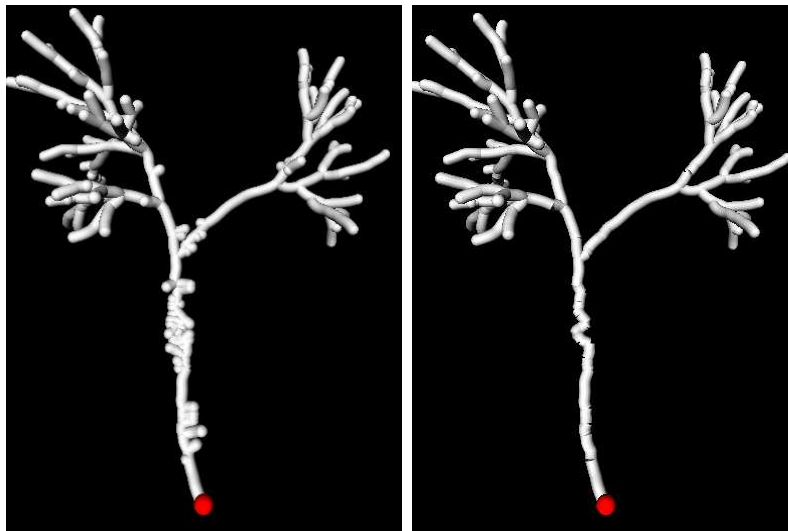


Abbildung 3.1.: Pruning eines Bronchialbaums

Dargestellt ist derselbe Bronchialbaum. In der linken Abbildung ist der Bronchialbaum vor dem Pruning dargestellt, rechts danach. Deutlich zu Erkennen ist das Entfernen vieler kurzer (anatomisch nicht vorhandener) Kanten am eingehenden Gefäß.

---

<sup>3</sup> $\text{pred}(v)$  sei der Vorgänger von  $v$ .

### Transformation in ein gemeinsames Koordinatensystem

Ein weitere wünschenswerte Eigenschaft für die Bäume wäre, dass zu matchende Knoten ähnliche Koordinaten besitzen sollten. Dies ist a priori nicht garantiert, da sich durch Lageveränderungen des Patienten, Atmung und Herzschlag die räumliche Position der Organe erheblich verändern kann. Somit müssten die Bäume nachträglich in ein gemeinsames Koordinatensystem verschoben werden. Dazu wären verschiedene Ansätze denkbar:

- Für Bronchialbäume ist ein gangbarer Weg der in [TMP<sup>+</sup>05] skizzierte Ansatz. Hierbei werden zunächst einige *Hauptverzweigungsknoten*<sup>4</sup> der Bronchialbäume detektiert und einander zugeordnet. Aus diesen Zuordnungen lässt sich – zum Beispiel mittels eines least-squares Ansatzes wie in [HHN88] – eine affine Registrierung ableiten, die den korrespondierenden Hauptverzweigungsknoten möglichst ähnliche Koordinaten zuordnet. Die Detektion von Hauptverzweigungsknoten ist in den Bronchialbäumen relativ stabil, dies ist jedoch bei den Gefäßsystemen der Leber nicht gegeben. Die Ursache hierfür ist wiederum, dass es sich bei dem Bronchialbaum um dichotome Strukturen handelt, bei den Gefäßsystemen der Leber dagegen nicht. Daher wurde dieser Ansatz nicht weiter verfolgt.
- Ein weiterer Ansatzpunkt wäre eine rigide Vor-Registrierung der zugrunde liegenden Bilddaten, mittels eines Ansatzes über *Mutual Information*[WVA<sup>+</sup>96]. Allerdings wäre hierzu wieder ein Rückschritt auf die Ebene der Bilddaten notwendig und das Finden einer sinnvollen rigiden Vor-Registrierung erscheint schwierig.

Die oben skizzierten Schwierigkeiten legen nahe, dass es sich bei der Transformation in ein gemeinsames Koordinatensystem um ein ähnlich schweres Problem wie bei dem eigentlichen Matching-Problem handelt. Daher wurde darauf verzichtet, die Bäume in ein gemeinsames Koordinatensystem zu transformieren. Dies impliziert, dass den absoluten Koordinaten von Knoten im weiteren keine Bedeutung beigemessen werden darf. Dies ist akzeptabel, weil Eigenschaften wie zum Beispiel die Entfernungen zwischen zwei Knoten relative Maße sind, die invariant gegenüber rigiden Transformationen sind.

## 3.2. Constraints zur Erzeugung des Assoziationsgraphen

Im Folgenden wird eine Reihe von Constraints vorgestellt. Diese sind im Gegensatz zu den Kapitel 2.3.2 vorgestellten Constraints speziell auf die gegebene Aufgabenstellung ausgerichtet. Die Qualität des Matchings hängt wesentlich von einer Wahl

---

<sup>4</sup>Zur Erläuterung des Begriffs "Hauptverzweigungsknoten" sei auf Abschnitt 3.2.4 verwiesen.

geeigneter Constraints ab. Daher werden die in diesem Abschnitt vorgestellten Constraints in Abschnitt 5.2.2 validiert.

### 3.2.1. Unäre Constraints

Unäre Constraints haben das Ziel, möglichst früh im Matching-Prozess solche Knotenpaare zu identifizieren, die aufgrund stark differierender Eigenschaften keine sinnvollen Zuordnungen darstellen. Dies hat den Vorteil, dass der Assoziationsgraph relativ klein gehalten werden kann, was die Laufzeit des Verfahrens deutlich verbessern kann. Das Auffinden einer maximalen Clique in kleinen Assoziationsgraphen ist außerdem deutlich einfacher.

Eine nahe liegende Idee für unäre Constraints ist es, nur solche Knoten zu matchen, die vergleichsweise ähnliche Eigenschaften besitzen. Solche lokale Eigenschaften sind beispielsweise der Gefäßdurchmesser oder der Verzweigungswinkel zwischen zwei Gefäßabschnitten.

Ist die Wurzel der Bäume sinnvoll gewählt – was bei den Eingabedaten in aller Regel der Fall ist – so wird durch die Baumhierarchie zudem eine Zuordnungen eines Teilbaums zu dem ihn induzierenden Knoten möglich. Dies gibt Anlass für zwei weitere Constraints: So sollten die von zwei zu matchenden Knoten induzierten Teilbäume möglichst ähnlich sein.

Schließlich gibt es in den Bäumen einige ausgezeichnete Knoten, die relativ einfach zu identifizieren sind. Als Beispiel hierfür seien die Wurzeln sowie die ersten Hauptverzweigungsknoten der Bäume genannt. Ein solcher Knoten wird als *Referenzknoten* bezeichnet. Zu genaueren Erläuterungen sei auf Abschnitt 3.2.4 verwiesen. Die Wahl eines solchen Referenzknotens gibt Anlass für weitere unäre Constraint: So sollten gewisse Eigenschaften wie die relative Lage der Knoten zu den Referenzknoten ähnlich sein, damit eine Zuordnung sinnvoll ist. Die Namen der Constraints, die von der Wahl eines Referenzknotens abhängen, sind durch das Präfix RN<sup>5</sup> gekennzeichnet.

Für diese drei Gruppen von Constraints werden im weiteren einige Beispiele vorgestellt, die jeweils auch implementiert wurden. Die folgende Menge an Constraints erhebt jedoch keinen Anspruch auf Vollständigkeit; es sind noch eine Reihe weiterer Constraints denkbar.

#### BifurcationAngle – Constraint

Im BifurcationAngle – Constraint wird ein Ähnlichkeitsmaß für Bifurkationsknoten – also solchen Knoten mit genau zwei Kindknoten – definiert. Jedem Bifurkationsknoten  $v$  ist durch seine beiden ausgehenden Kanten ein *Verzweigungswinkel*  $\phi[v]$  zugeordnet. Für zwei Bifurkationsknoten  $v_1, v_2$  aus verschiedenen Bäumen, die zugeordnet werden sollen, sollten diese Winkel möglichst ähnlich sein. Das Verhältnis  $\phi[v_1]/\phi[v_2]$  sollte also ungefähr 1 sein. Aus diesem Verhältnis lässt sich mittels

---

<sup>5</sup>RN wie Reference Node.



der Bewertungsfunktion  $a_\sigma$  aus Abschnitt 3.2.3 ein Ähnlichkeitsmaß definieren. Ist entweder  $v_1$  oder  $v_2$  kein Bifurkationsknoten, so wird die neutrale Bewertung 0.5 gegeben.

### EdgeDiameter – Constraint

Jeder Knoten – mit Ausnahme der Wurzel – besitzt nach der Vorverarbeitung aus Abschnitt 3.1 eine eindeutige eingehende Kante. Dieser Kante sind gewisse Parameter wie Länge, Volumen oder Durchmesser des korrespondierenden Gefäßabschnitts zugeordnet. Idealerweise sollten für zwei zu matchende Knoten obige Eigenschaften der eingehenden Kanten sehr ähnlich sein. Durch Segmentierungsprobleme, die z. B. zusätzliche Knoten bewirken und eine Kante zwei teilen können (vergleiche Abbildung 3.6), können Parameter wie Länge und Volumen jedoch fälschlicherweise stark differieren. Daher sind diese als Ähnlichkeitsmaß leider ungeeignet. Anders verhält es sich jedoch mit dem *mittleren Durchmesser* einer Kante. Dieser Parameter, der in etwa dem Verhältnis von Volumen zu Länge entspricht, sollte vergleichsweise unempfindlich gegenüber obigen Segmentierungsproblemen sein. Daher scheint es sinnvoll, zu fordern, dass für zwei zu matchende Knoten das Verhältnis der mittleren Durchmesser der eingehenden Kanten ungefähr 1 ist.

### SubtreeSimilarity – Constraint

Im SubtreeSimilarity – Constraint wird die *Größe* eines von einem Knoten  $v$  induzierten Teilbaums  $size(T[v])$  benötigt. Unter der Größe eines Teilbaums verstehen wir die Summe der Länge aller Kanten in diesem Baum.

In vielen Situationen ist folgende Bedingung für ein Matching zweier Knoten  $v_1 \leftrightarrow v_2$  sinnvoll: Die Größen  $size(T[v_1])$  und  $size(T[v_2])$  sollten in etwa gleich groß sein. Dies erscheint angemessen, wenn sich die jeweiligen anatomischen Teilbäume zwar in Lage und Orientierung ändern, die Länge der einzelnen Gefäßabschnitte jedoch in etwa gleich bleibt. Das ist immer dann der Fall, wenn die Anatomie der zugrunde liegenden Organe sich kaum verändert. Als Gegenbeispiel sei die Leberregeneration genannt, bei der es zu großen Änderungen des Gefäßsystems kommen kann: So kann beispielsweise ein an einem Bifurkationsknoten  $v$  wurzelnder Teilbaum im Laufe der Regeneration seine Größe erheblich steigern.

Außerdem kann es gelegentlich dazu kommen, dass durch einen Segmentierungsfehler ein anatomischer Teilbaum nicht gefunden wurde, sodass dieser in den Eingabedaten fehlt. Daraus resultiert eine Veränderung der Größe aller Bäume, die diesen Teilbaum enthalten hätten.

Ein weiteres Problem ist die verschiedene Auflösung der bildgebenden Verfahren. So können beim CT-Verfahren noch feinere Gefäßsysteme erfasst werden als beim MRT-Verfahren, sodass der Baum aus mehr Knoten und Kanten besteht. Daraus resultiert eine Änderung der Größe der Bäume. Dieses Problem lässt sich aber mindern, wenn man statt der absoluten die relativen<sup>6</sup> Größen der Teilbäume vergleicht,

<sup>6</sup>Relativ bezüglich der Größe des Gesamtbaums.

also:

Für eine Zuordnung  $v_1 \leftrightarrow v_2$  sollten die Größen  $\frac{size(T[v_1])}{size(T_1)}$  und  $\frac{size(T[v_2])}{size(T_2)}$  in etwa gleich groß sein. Hierbei sei  $T_i$  jeweils der Baum, in dem sich  $v_i$  befindet.

Das Verhältnis dieser beiden Größen sollte also nahe 1 liegen. Aus diesem Verhältnis lässt sich mittels der Bewertungsfunktion  $a_\sigma$  aus Abschnitt 3.2.3 ein Ähnlichkeitsmaß definieren.

### BifurcationSimilarity – Constraint

Analog zum vorhergehenden Constraint, wird auch in diesem Constraint wieder die Größe eines Teilbaums verwendet. Daraus ergibt sich, dass auch dieser Constraint nicht im Falle von anatomischen Veränderungen der Organe, wie der Leberregeneration, sinnvoll ist.

Dieser Constraint ist weiterhin nur für *Bifurkationsknoten* sinnvoll, also solche Knoten, die genau zwei Kindknoten haben, in denen sich das Gefäßsystem somit verzweigt.

Ein wichtiges Unterscheidungskriterium für solche Bifurkationsknoten ist die Art der Verzweigung des Gefäßsystems. Beispielsweise könnte sich das Gefäßsystem in zwei etwa gleich große Systeme verzweigen oder aber in einen großen Hauptast und einen kleinen Nebenast. Daher erscheint es sinnvoll, dass bei einem Matching zweier Bifurkationsknoten  $v_1 \leftrightarrow v_2$  die Art der Verzweigung gleich sein sollte.

Dazu wird zunächst das Verhältnis der Größen der beiden von einem Bifurkationsknoten  $v$  ausgehenden Gefäßsysteme berechnet. Dieses sei  $split(v) = (size(T[v_1]) + d(e_1)) / (size(T[v_2]) + d(e_2))$ . Hierbei seien  $v_1$  und  $v_2$  die beiden Kindknoten von  $v$  und  $d(e_i)$  die Länge der in  $v_i$  eingehenden Kante. Ohne Beschränkung der Allgemeinheit seien die Indizes der Kindknoten so gewählt, dass  $split(v) \geq 1$  sei.

Damit lässt sich der BifurcationSimilarity – Constraint wie folgt definieren: Für zwei zu matchende Bifurkationsknoten  $v_1 \leftrightarrow v_2$  sollten  $split(v_1)$  und  $split(v_2)$  in etwa gleich groß sein; das Verhältnis dieser beiden Größen sollte also nahe 1 liegen. Aus diesem Verhältnis lässt sich wiederum mittels der Bewertungsfunktion  $a_\sigma$  aus Abschnitt 3.2.3 ein Ähnlichkeitsmaß definieren.

### RN-PathCurvature – Constraint

Eine weitere charakteristische Eigenschaft eines Knotens  $v$  ist die *Geradlinigkeit* bzw. *Krümmung*  $curv()$  seines Pfads zu einem Referenzknoten  $r$ . Unter Krümmung eines Pfads verstehen wir hierbei das Verhältnis der Pfadlänge  $d_p$ <sup>7</sup> zur Distanz seiner Endknoten:

$$curv(v, r) = \frac{d_p(v, r)}{\|v - r\|_2}$$

---

<sup>7</sup>Die Pfadlänge ist definiert als die Summe der Länge der Kanten auf dem eindeutigen Pfad im Baum zwischen zwei Knoten. Die Länge wird während des Baumextraktionsverfahren gemessen und als Attribut in der Kante hinterlegt.

Ist  $curv(v, r) \approx 1$ , so ist der Pfad sehr geradlinig, wohingegen  $curv(v, r) \gg 1$  bedeutet, dass der Pfad vergleichsweise gekrümmt. Unter der Annahme, dass die Referenzknoten  $r_1$  und  $r_2$  in beiden Bäumen sinnvoll gewählt wurden, sollte für das Matching zweier Knoten  $v_1 \leftrightarrow v_2$  gelten, dass  $curv(v_1, r_1) \approx curv(v_2, r_2)$  ist. Das Verhältnis  $r = \frac{curv(v_1, r_1)}{curv(v_2, r_2)}$  ist somit ein gutes Maß für die Konsistenz der Zuordnungen  $v_1 \leftrightarrow v_2$ . Hierbei gilt wiederum, dass ein Matching zweier Knoten umso sinnvoller ist, je näher  $r$  an 1 liegt.

### RN-CurveSimilarity – Constraint

Der RN-PathCurvature – Constraint gibt ein qualitatives Maß für die Ähnlichkeit zweier Pfade an. Vorteilhaft ist hierbei, dass sich dieses Maß sehr schnell berechnen lässt. Nachteilhaft ist hingegen, dass zwei Pfade, die sehr unterschiedliche Form haben, eine ähnliche Krümmung besitzen können. Während bei dem blauen Pfad in Abbildung 3.2 der gewundene Teil nahe dem Referenzknoten  $r$  ist und der Pfad nahe dem Knoten  $v$  recht geradlinig verläuft, verhält es sich bei dem roten Pfad genau umgekehrt. Dennoch ist  $curv(v, r)$  jeweils gleich.

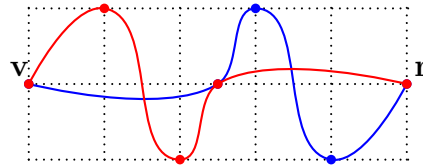


Abbildung 3.2.: Zwei verschiedene Pfade mit ähnlicher Krümmung

Schematisch dargestellt sind die räumlichen Verläufe der Pfade von zwei Knoten  $v$  zu ihren Referenzknoten  $r$ . Der rot gefärbte Pfad ist nahe dem Ausgangsknoten stark gewunden während der blaue Pfad nahe dem Referenzknoten stark gewunden ist. Dennoch werden diese beiden Pfade vom RN-PathCurvature – Constraint als ähnlich bewertet.

Deswegen ist es sinnvoll, zusätzlich ein genaueres Maß für die Ähnlichkeit des räumlichen Verlaufs zweier Pfade einzuführen. Dieses wird im folgenden vorgestellt. Dazu seien die eindeutigen Pfade vom jeweiligen Knoten  $v$  zu den Referenzknoten  $r$  in Form einer Folge von Koordinaten auf diesem Pfad gegeben. Diese Folge von Koordinaten definiert einen Streckenzug  $S = [v, v_1, \dots, v_n, r]$  im  $\mathbb{R}^3$ . Im Weiteren wird ein Verfahren entwickelt, um die Ähnlichkeit zweier solcher Streckenzüge zu ermitteln. Die Länge der Streckenzüge soll dabei keinen direkten Einfluss auf die Ähnlichkeit haben, wohl aber die Form des Streckenzuges. Dazu müssen diese zunächst "normiert" werden; die Streckenzüge werden dazu wie folgt transformiert:

1. Der Streckenzug wird um  $-v$  verschoben, sodass der Startpunkt des Streckenzuges im Ursprung zu liegen kommt. Dann gilt  $S' = [0, v_1 - v, \dots, v_n - v, r - v]$ .

2. Der Streckenzug  $S'$  wird so skaliert, dass Start- und Endpunkt die euklidische Distanz 1 haben. Dazu wird der Ortsvektor jedes Punktes auf dem Streckenzug mit  $\frac{1}{\|r-v\|_2}$  multipliziert.
3. Der Streckenzug wird so rotiert, dass sein Endpunkt auf dem Punkt  $x_1 = (1, 0, 0)^t$  zu liegen kommt. Dazu wird zunächst der Winkel  $\alpha$  zwischen den Vektoren  $r - v$  und der ersten Koordinatenachse  $x$  berechnet. Dann wird der Streckenzug um  $\alpha$  um die zu  $r - v$  und  $x$  orthogonale Achse rotiert<sup>8</sup>.

Durch diese Transformation sind die Streckenzüge derart normiert worden, dass sie alle in  $s_0 = (0, 0, 0)^t$  beginnen und in  $s_n = (1, 0, 0)^t$  enden. Sei dieser transformierte Streckenzug  $S = [s_0, s_1, \dots, s_{n-1}, s_n]$ . Weiterhin bezeichne  $S_i = [s_0, s_1, \dots, s_i]$  einen Teil-Streckenzug und  $\|S_i\|$  die Länge eines Streckenzuges, also

$$\|S_i\| = \sum_{j=1}^i \|s_j - s_{j-1}\|_2.$$

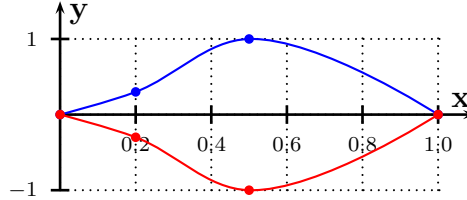


Abbildung 3.3.: Zwei zu vergleichende, normierte Streckenzüge

Wird der Rotationsfreiheitsgrad bezüglich der x-Achse nicht berücksichtigt, so kann es auch bei ähnlicher Form zweier Streckenzüge zu großen Abständen kommen.

Nun wird eine Parametrisierung des transformierten Streckenzuges  $S$  eingeführt. Hierunter ist eine Funktion  $f : [0; 1] \rightarrow \mathbb{R}^3$  zu verstehen mit  $f(\frac{\|S_i\|}{\|S\|}) = s_i$ , insbesondere also  $f(0) = (0, 0, 0)^t$  und  $f(1) = (1, 0, 0)^t$ . Die anderen Werte von  $f$  seien linear interpoliert:

Für  $t \in \left[ \frac{\|S_i\|}{\|S\|}, \frac{\|S_{i+1}\|}{\|S\|} \right]$  gelte  $f(t) = (1 - \alpha)s_i + \alpha s_{i+1}$  mit  $\alpha = \frac{t\|S\| - \|S_i\|}{\|S_{i+1}\| - \|S_i\|}$

Offenbar ist für zwei Streckenzüge  $S_1$  und  $S_2$  mit Parametrisierungen  $f_1$  und  $f_2$  das Integral  $\int_0^1 \|f_1(t) - f_2(t)\|_2 dt$  eine Art Ähnlichkeitsmaß. Anschaulich gibt das Integral die Fläche zwischen den beiden Streckenzügen an. Wie in Abbildung 3.3 zu sehen ist, kann es bei diesem Ähnlichkeitsmaß allerdings zu großen Werten kommen, obwohl zwei Kurven die gleiche Form besitzen. Die Ursache ist, dass zwar Start- und Endpunkt der Streckenzüge angeglichen wurden, jedoch ein Freiheitsgrad bezüglich der Rotation um die x-Achse verbleibt. Im Beispiel würde eine Rotation von einer der beiden Kurven um  $180^\circ$  um die x-Achse den Wert des Integrals auf 0 senken.

<sup>8</sup>Diese Achse ist eindeutig bestimmt, sofern nicht  $r - v = (+/- 1, 0, 0)^t$  ist. In diesem Fall kann eine zu  $x$  orthogonale Achse frei gewählt werden, weil  $\alpha = 0$  oder  $\alpha = \pi$  gilt.

Ein sinnvolleres Ähnlichkeitsmaß ist somit der Wert des folgenden Minimierungsproblems:

$$d = \min_{\phi \in [0, 2\pi]} d(\phi) \text{ mit } d(\phi) = \int_0^1 \|f_1(t) - A_\phi f_2(t)\|_2 dt \quad (3.1)$$

Bei  $A_\phi$  handelt es um eine Rotationsmatrix, die eine Drehung um  $\phi$  um die x-Achse beschreibt. Diese sind von der Form:

$$A_\phi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix}$$

Als schwierig erweist sich in diesem Fall die Bestimmung des optimalen Winkels  $\phi$ . Ein Ansatz ist, den Drehwinkel zu diskretisieren und den Wert des Integrals für alle  $\phi \in \left\{0, \frac{2\pi}{n}, \dots, \frac{2\pi(n-2)}{n}, \frac{2\pi(n-1)}{n}\right\}$  zu berechnen. Bei hinreichend großem  $n$  sollte ein Wert nahe am globalen Minimum gefunden werden. Allerdings steigt mit  $n$  auch die Anzahl der zu berechnenden Integrale, was die Rechenzeit erheblich erhöht. Alternativ kann auch eine binäre Suche durchgeführt werden. In dieser wird bei Eingabe eines geeigneten Suchintervalls<sup>9</sup> ein lokales Minimum gefunden und die Anzahl der zu berechnenden Integrale ist vergleichsweise klein. Das Verfahren ist in Prozedur BinarySearch skizziert.

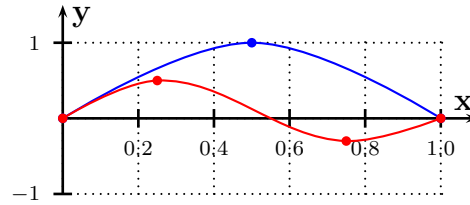


Abbildung 3.4.: Zwei Streckenzüge mit zwei lokalen Minima für  $\phi$

Die Drehwinkel 0 und  $\pi$  sind beide lokale Minima der Minimierung von  $d$  bezüglich  $\phi$ . Die Konvergenz der binären Suche gegen das globale Minimum ( $\phi = 0$ ) ist nicht garantiert.

In der Regel existiert nur ein lokales Minimum, sodass die binäre Suche gegen das globale Minimum konvergiert; es gibt jedoch auch Ausnahmen, wie die Streckenzüge in Abbildung 3.4. Dort sind die Drehwinkel 0 und  $\pi$  beide lokale Minima. Im Allgemeinen liefert die binäre Suche jedoch bessere Ergebnisse bei weniger Rechenaufwand als das Durchprobieren vieler Werte für  $\phi$ . Alternativ ließen sich auch

<sup>9</sup>Das Intervall  $I = [a, b]$  muss dazu ein lokales Minimum beinhalten. Die Konvergenz des Verfahrens ist gesichert, wenn gilt:  $\frac{d}{d\phi}d(\phi)|_{\phi=a} < 0$  und  $\frac{d}{d\phi}d(\phi)|_{\phi=b} > 0$ .

Ein solches Intervall ist, sofern  $d(\phi)$  nicht auf  $[0, 2\pi]$  konstant ist, leicht zu finden. Auf eine ausführlichere Darstellung wird der Einfachheit halber verzichtet.

```

/* Wähle ein geeignetes Intervall  $I = [a, b]$  wie oben erläutert */
 $I = [a, b]$ ;
while true do
    /* Falls lokales Minimum hinreichend gut eingegrenzt ist */
    if  $b - a < \frac{2\pi}{k}$  then
        /* Gebe Approximation des lokalen Minimums zurück */
        return  $d = d(\frac{a-b}{2})$ ;
    else
        /* Berechne Gradient in der Mitte des Intervalls */
         $t \leftarrow \left[ \frac{d}{d\phi} d(\phi) \Big|_{\phi=\frac{a-b}{2}} \approx \frac{d(\frac{a-b}{2}+\epsilon) - d(\frac{a-b}{2}-\epsilon)}{2\epsilon} \right]$ ;
        /* Wenn Gradient größer Null  $\Rightarrow$  In der linken
           Intervallhälfte ist ein lokales Minimum */
        if  $t > 0$  then
             $b \leftarrow \frac{a-b}{2}$ ;
        /* Sonst: In der rechten Intervallhälfte ist ein lokales
           Minimum */
        else
             $a \leftarrow \frac{a-b}{2}$ ;

```

#### Prozedur BinarySearch

Verfahren mit besserer Konvergenz wie zum Beispiel das *Newton-Verfahren* einsetzen.

Die Kurven sind umso ähnlicher, je kleiner  $d$  ist, wobei natürlich  $d \geq 0$  ist. Somit sollte  $1+d$  möglichst nahe an 1 liegen. Mittels der Bewertungsfunktion aus Abschnitt 3.2.3 angewendet auf  $1+d$  kann ein Ähnlichkeitsmaß kompatibel zu denen der anderen Constraints definiert werden.

#### RN-CurveTangent – Constraint

Der RN-CurveSimilarity – Constraint liefert zwar im Allgemeinen ein sinnvolles Ähnlichkeitsmaß für zwei Streckenzüge, jedoch gibt es Fälle, bei denen Paare von Streckenzügen erheblich differierende Ähnlichkeits-Bewertungen erhalten, obwohl eigentlich ähnliche Bewertungen wünschenswert wären. So werden in Abbildung 3.5 die beiden Streckenzüge im linken Beispiels als deutlich ähnlicher eingestuft als die im rechten Beispiel. Hier wäre es jedoch bei beiden Paaren von Streckenzügen sinnvoll gewesen, sie als ähnlich einzuordnen, weil sie jeweils größtenteils parallel verlaufen.

Das Problem liegt darin, dass als Grundlage für die Ähnlichkeit der Kurven die absoluten Koordinaten der Streckenzüge benutzt werden. Alternativ könnten auch die Richtungen der Kurven als Ausgangspunkt gewählt werden. Da die Streckenzüge stückweise linear sind, existieren die Ableitungen  $f'(t)$  fast überall und  $f'$  ist

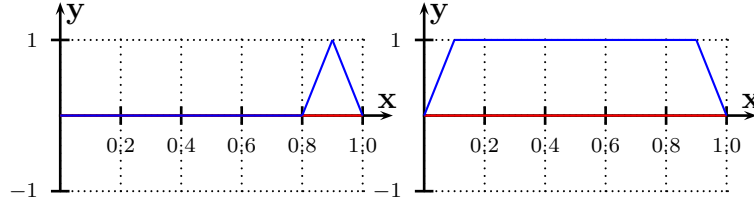


Abbildung 3.5.: Beispiel für unintuitive Bewertung der Ähnlichkeit zweier Kurven

Die beiden Streckenzüge in der linken Abbildung werden vom RN-CurveSimilarity – Constraint als deutlich ähnlicher eingestuft als die in der rechten Abbildung. Es wäre jedoch bei beiden Paaren von Streckenzügen sinnvoll gewesen, sie als ähnlich einzuordnen, weil sie jeweils größtenteils parallel verlaufen.

stückweise konstant<sup>10</sup>. Bei  $g_i(t) = \frac{f'_i(t)}{\|f'_i(t)\|_2}$  handelt es sich um Einheitsvektoren, die die Richtung der Tangenten des Streckenzuges für Parameterwert  $t$  angeben. Die Ähnlichkeit zweier normierter Vektoren  $a$  und  $b$  (in diesem Fall also der Richtungsvektoren) lässt sich mittels des Skalarprodukts bestimmen. Es gilt:  $a^T b \leq 1$  und  $a^T b = 1 \Leftrightarrow a = b$ . Das Integral  $\int_0^1 (1 - g_1(t)^T g_2(t)) dt$  ist also stets  $\geq 0$  und die Richtungen der Streckenzüge sind umso ähnlicher je näher der Wert an Null liegt. Wie im vorhergehenden Constraint sind  $g_1(t)$  und  $g_2(t)$  jedoch nicht unmittelbar vergleichbar, da der Rotationsfreiheitsgrad um die x-Achse auch die Tangentenrichtungen beeinflusst. Daher ist es sinnvoll, statt des Skalarprodukts  $g_1(t)^T g_2(t)$  den minimalen Wert (bezüglich  $A$ ) von  $g_1(t)^T A g_2(t)$  zu bestimmen. Hierbei bezeichne  $A$  eine orthogonale Matrix<sup>11</sup>, also  $AA^T = 1$ .

Zusammenfassend: Gegeben sind zwei Funktionen  $g_1, g_2 : [0, 1] \rightarrow S^2$ . Hierbei handelt es sich um die normierten Tangentialvektoren der parametrisierten Streckenzüge  $f_1(t)$  und  $f_2(t)$ .  $g_1$  und  $g_2$  sind stückweise konstant, also integrierbar. Zu lösen ist nun das Optimierungsproblem:

$$\min_{A: AA^T=1} \int_0^1 (1 - g_1(t)^T A g_2(t)) dt \quad (3.2)$$

Es gilt:

$$\begin{aligned} \min_{A: AA^T=1} \int_0^1 (1 - g_1(t)^T A g_2(t)) dt &= 1 - \max_{A: AA^T=1} \int_0^1 g_1(t)^T A g_2(t) dt \\ &= 1 - \max_{A: AA^T=1} \int_0^1 \sum_{j=1}^3 \sum_{k=1}^3 g_{1j}(t) a_{jk} g_{2k}(t) dt \end{aligned}$$

<sup>10</sup>An den Stellen, wo  $f'(t)$  nicht existiert, kann es durch  $f'(t - \epsilon)$  oder  $f'(t + \epsilon)$  ergänzt werden.

<sup>11</sup>Es ist sinnvoll für  $A$  nur orthogonale Matrizen zuzulassen, da wir ja explizit eine Optimierung bezüglich einer Rotation um die x-Achse erreichen wollen.

$$\begin{aligned}
&= 1 - \max_{A: AA^T=1} \sum_{j=1}^3 \sum_{k=1}^3 \int_0^1 g_{1j}(t) a_{jk} g_{2k}(t) dt \\
&= 1 - \max_{A: AA^T=1} \sum_{j=1}^3 \sum_{k=1}^3 a_{jk} b_{jk}
\end{aligned}$$

Hierbei wurde  $g_i = (g_{i1}, g_{i2}, g_{i3})^T$  und  $b_{jk} = \int_0^1 g_{1j}(t) g_{2k}(t) dt$  gesetzt.

Um obigen Ausdruck noch weiter zu vereinfachen, müssen zusätzliche Anforderungen an die Matrix  $A$  gestellt werden. Nach Konstruktion der beiden Streckenzüge  $S_1$  und  $S_2$  gilt für die Hauptrichtungen  $r_1$  und  $r_2$  der Streckenzüge:  $r_1 = r_2 = (1, 0, 0)^T$ . Eine sinnvolle zusätzliche Anforderung an die Matrix  $A$  ist:  $r_1^T A r_2 = 1 \Leftrightarrow A r_2 = r_1$ . Aus dieser Bedingung und  $AA^T = 1$  folgt, dass es sich bei  $A$  um eine Rotationsmatrix um die erste Koordinatenachse handeln muss, also:

$$A = A_\phi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix}$$

Wie beim RN-CurveSimilarity – Constraint beschränken wir uns also auf eine Optimierung bezüglich der Rotation um die x-Achse. Nun lässt sich das Problem 3.2 weiter vereinfachen:

$$\begin{aligned}
\min_{A_\phi} \int_0^1 (1 - g_1(t)^T A g_2(t)) dt &= 1 - \max_{A_\phi} \sum_{j=1}^3 \sum_{k=1}^3 a_{jk} b_{jk} \\
&= 1 - \max_{\phi \in [0, 2\pi]} (b_{11} + (b_{22} + b_{33}) \cos(\phi) + (b_{23} - b_{32}) \sin(\phi)) \\
&= 1 - C - \max_{\phi \in [0, 2\pi]} (D \cos(\phi) + E \sin(\phi))
\end{aligned}$$

wobei  $C = b_{11}$ ,  $D = b_{22} + b_{33}$  und  $E = b_{23} - b_{32}$  gesetzt wurde. Es gilt:

$$\begin{aligned}
\max_{\phi \in [0, 2\pi]} (D \cos(\phi) + E \sin(\phi)) &= \sqrt{D^2 + E^2} \max_{\phi \in [0, 2\pi]} \left( \cos(\phi) \frac{D}{\sqrt{D^2 + E^2}} + \sin(\phi) \frac{E}{\sqrt{D^2 + E^2}} \right) \quad (1) \\
&= \sqrt{D^2 + E^2} \max_{\phi \in [0, 2\pi]} (\cos(\phi) \sin(\beta) + \sin(\phi) \cos(\beta)) \\
&= \sqrt{D^2 + E^2} \max_{\phi \in [0, 2\pi]} \sin(\phi + \beta) \quad (2) \\
&= \sqrt{D^2 + E^2}
\end{aligned}$$

Hierbei wurde bei (1) ausgenutzt, dass wegen  $\left(\frac{D}{\sqrt{D^2 + E^2}}\right)^2 + \left(\frac{E}{\sqrt{D^2 + E^2}}\right)^2 = 1$  ein  $\beta \in [0, 2\pi]$  existiert mit  $\sin(\beta) = \frac{D}{\sqrt{D^2 + E^2}}$  und  $\cos(\beta) = \frac{E}{\sqrt{D^2 + E^2}}$ . Bei (2) wurde benutzt, dass  $\sin(\phi + \beta)$   $2\pi$ -periodisch in  $\phi$  ist und  $\max_{\phi \in [0, 2\pi]} \sin(\phi + \beta) = 1$  gilt.

Es erübrigt sich also eine Berechnung von  $\phi$ . Die Lösung des Problems 3.2 ist:

$$\begin{aligned}
&\min_{A_\phi} \int_0^1 (1 - g_1(t)^T A_\phi g_2(t)) dt = 1 - C - \sqrt{D^2 + E^2} \\
&\text{mit } C = b_{11}, D = b_{22} + b_{33}, E = b_{23} - b_{32} \text{ und } b_{jk} = \int_0^1 g_{1j}(t) g_{2k}(t) dt.
\end{aligned}$$



### 3.2.2. Binäre Constraints

In diesem Abschnitt wird eine Reihe binärer Constraints vorgestellt. Ein solcher binärer Constraint gibt an, wann zwei Knoten im Assoziationsgraphen miteinander verbunden werden sollen – wann also zwei Zuordnungen von Knoten miteinander konsistent sind. In Abschnitt 2.3.2 wurden zwei binäre Constraints angegeben, die im Wesentlichen auf der topologischen Distanz innerhalb von Bäumen beruhen.

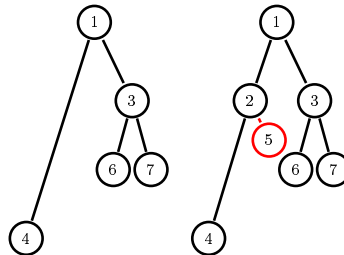


Abbildung 3.6.: Anfälligkeit der topologischen Distanz für Segmentierungsartefakte:

Die topologische Distanz des Knotens 4 zu jedem der anderen Knoten ist in beiden Bäumen verschieden, obwohl im linken Baum nur die relativ kurze und nicht signifikante Kante zu Knoten 5 übersehen wurde. Die Pfadlänge  $d_p$  ist hingegen robust gegen Rauschen und Segmentierungsartefakte.

Bei Baumdaten, die aus Segmentierungsverfahren resultieren, ist die topologische Distanz kein geeigneter Ansatzpunkt, da sie nicht robust gegenüber fehlenden Verzweigungen im Baum ist, die fälschlicherweise nicht segmentiert wurden. So ändert sich in Abbildung 3.6 die topologische Distanz des Knotens 4 zu allen anderen Knoten, obwohl im linken Baum nur die relativ kurze und nicht signifikante Kante zu Knoten 5 übersehen wurde.

Statt der topologischen Distanz werden im weiteren einige robustere Ähnlichkeitsmaße vorgestellt.

#### PathLength – Constraint

Da jeder Kante im Baum eine Länge zugeordnet ist, ist eine nahe liegende Alternative zur topologischen Distanz die *Pfadlänge*  $d_p$ <sup>12</sup>. Offenbar ist sie robust gegen Rauschen und Segmentierungsartefakte wie in Abbildung 3.6. So ändert sich im rechten Baum zwar die Anzahl Kanten auf dem Pfad von Knoten 4 zur Wurzel, die Pfadlänge bleibt jedoch gleich, da die Länge der Kante (4, 1) im linken Baum gleich der Summe der Längen der Kanten (4, 2) und (2, 1) im rechten Baum ist.

<sup>12</sup>Die Pfadlänge ist wiederum definiert als die Summe der Länge der Kanten auf dem eindeutigen Pfad im Baum zwischen zwei Knoten. Die Länge wird während des Baumextraktionsverfahren gemessen und als Attribut in der Kante hinterlegt.

Als ein Maß für die Konsistenz der Knotenzuordnungen  $v_{A1} \leftrightarrow v_{A2}$  und  $v_{B1} \leftrightarrow v_{B2}$  kann das Verhältnis  $r = \frac{d_p(v_{A1}, v_{B1})}{d_p(v_{A2}, v_{B2})}$  angesehen werden. Je näher das Verhältnis an 1 liegt, desto verträglicher sind die Zuordnungen.  $r = 1$  bedeutet anschaulich, dass die gematchten Knoten in den beiden Bäumen durch gleich lange Pfade verbunden sind.

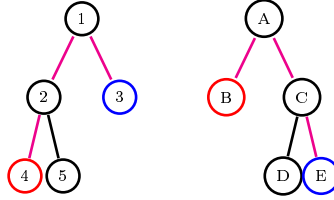


Abbildung 3.7.: Probleme bei ungerichteter Pfadlänge

Die rot gefärbte Zuordnung ist mit der blau gefärbten Zuordnung bezüglich der Pfadlänge konsistent. Dies erscheint jedoch unintuitiv; die Ursache hierfür ist, dass der Knoten  $B$  deutlich näher an der Wurzel liegt, als der Knoten 4. Daher sollte die gerichtete Pfadlänge als Kriterium verwendet werden.

In Bäumen mit Wurzeln ist die Pfadlänge jedoch kein optimales Kriterium. Die Zuordnungen  $4 \leftrightarrow B$  und  $3 \leftrightarrow E$  in Abbildung 3.7 sind bezüglich der Pfadlänge konsistent. Dennoch scheinen die beiden Zuordnungen hier nicht verträglich zu sein. Dies liegt daran, dass der Knoten  $B$  deutlich näher an der Wurzel liegt, als der Knoten 4. Es macht daher in Bäumen mit einer ausgezeichneten Wurzel Sinn, den Pfad in zwei Abschnitte zu unterteilen: Den Teil des Pfads von einem Knoten  $v_1$  zu einem Knoten  $v_2$ , der in Richtung der Wurzel verläuft, und den Teil, der sich von der Wurzel wieder in Richtung von  $v_2$  entfernt. Damit zwei Matchings verträglich sind, sollte die Länge beider Pfad-Abschnitte in etwa gleich sein, also:

$$r_1 = \frac{d_p(v_{A1}, lca(v_{A1}, v_{B1}))}{d_p(v_{A2}, lca(v_{A2}, v_{B2}))} \approx 1 \wedge r_2 = \frac{d_p(lca(v_{A1}, v_{B1}), v_{B1})}{d_p(lca(v_{A2}, v_{B2}), v_{B2})} \approx 1$$

Hierbei bezeichnet  $lca()$  den *least common ancestor*<sup>13</sup>.  $r_1$  ist damit das Verhältnis der beiden Pfadlängen aufwärts im Baum und  $r_2$  das Verhältnis der beiden Pfadlängen abwärts. Mit diesen beiden Bedingungen sind die Zuordnungen  $4 \leftrightarrow B$  und  $3 \leftrightarrow E$  in Abbildung 3.7 nicht mehr konsistent, was anschaulich auch sinnvoll erscheint.

<sup>13</sup>Als least common ancestor zweier Knoten  $v_1, v_2$  bezeichnet man den eindeutig bestimmten Knoten  $v$  auf dem Weg von  $v_1$  nach  $v_2$ , der minimales Level besitzt, vergleiche Appendix A auf Seite 71.

### PathCurvature – Constraint

Analog zum unären RN-PathCurvature – Constraint lässt sich auch ein binärer Constraint definieren. Sei dazu wieder die Krümmung eines Pfades als das Verhältnis der Pfadlänge zur Distanz seiner Endknoten definiert:

$$\text{curv}(v_1, v_2) = \frac{d_p(v_1, v_2)}{\|v_1 - v_2\|_2}$$

Ist  $\text{curv}(v_1, v_2) \approx 1$ , so ist der Pfad sehr geradlinig. Ist  $\text{curv}(v_1, v_2) \gg 1$ , so ist der Pfad dagegen vergleichsweise gekrümmt. Das Verhältnis  $r = \frac{\text{curv}(v_{A1}, v_{B1})}{\text{curv}(v_{A2}, v_{B2})}$  ist ein gutes Maß für die Konsistenz der Zuordnungen  $v_{A1} \leftrightarrow v_{A2}$  und  $v_{B1} \leftrightarrow v_{B2}$ . Hierbei gilt wiederum, dass zwei Zuordnungen umso konsistenter sind, je näher  $r$  an 1 liegt.

### CurveTangent – Constraint

Wie im Falle der unären Constraints ist es auch für die binären Constraints wünschenswert, dass es neben dem qualitativen Ähnlichkeitsmaß der Pfadkrümmung auch einen Constraint gibt, der ein exakteres Ähnlichkeitsmaß bildet. Sowohl aus dem RN-CurveSimilarity als auch aus dem RN-CurveTangent - Constraint lassen sich auf einfache Weise binäre Constraints ableiten. Werden im unären Fall die beiden Kurven von  $v_1$  bzw.  $v_2$  zu den Referenzknoten  $r_1$  bzw.  $r_2$  verglichen, so können – für die Überprüfung der Konsistenz der Zuordnungen  $v_{A1} \leftrightarrow v_{A2}$  und  $v_{B1} \leftrightarrow v_{B2}$  – einfach die Kurven von  $v_{A1}$  nach  $v_{B1}$  und von  $v_{A2}$  nach  $v_{B2}$  verglichen werden.

Zu beachten ist allerdings, dass die binären Constraints ungleich öfter als die unären Constraints geprüft werden müssen<sup>14</sup>. Aufgrund der höheren Laufzeit des CurveSimilarity – Constraint wurde auf diesen verzichtet und nur der CurveTangent – Constraint realisiert. Allerdings besitzt auch dieser eine vergleichsweise hohe Laufzeit (vergleiche Tabelle 5.3).

### Inheritance – Constraint

Während die topologische Distanz im Anwendungsfall kein geeignetes Ähnlichkeitsmaß ist, so gibt es doch gewisse andere topologische Eigenschaften, die zur Definition eines Constraints nützlich sind. Ist  $v_{A1}$  ein Vorfahre<sup>15</sup> von  $v_{B1}$ , so sollten die Zuordnungen  $v_{A1} \leftrightarrow v_{A2}$  und  $v_{B1} \leftrightarrow v_{B2}$  nur dann konsistent sein, wenn  $v_{A2}$  ebenfalls ein Vorfahre von  $v_{B2}$  ist. Mit anderen Worten: Das Matching soll die Vererbungshierarchien der beiden Bäume respektieren. Formal wird eine Funktion  $\text{inh} : V \times V \rightarrow \{\text{Ancestor}, \text{Descendant}, \text{Sibling}\}$  definiert. Dabei gelte:

<sup>14</sup>Unäre Constraints müssen genau für jeden potentiellen Knoten des Assoziationsgraphen einmal getestet werden, binäre Constraints dagegen genau einmal für jede potentielle Kante. Die Anzahl der Kanten ist quadratisch in der Anzahl der Knoten.

<sup>15</sup>Als Vorfahre eines Knotens  $v$  werden alle Knoten auf dem eindeutigen Pfad von  $v$  zur Wurzel (inklusive dieser) bezeichnet.

$$\text{inh}(v_1, v_2) = \begin{cases} \text{Ancestor} & \text{genau dann, wenn } v_1 \text{ ein Vorfahre von } v_2 \text{ ist} \\ \text{Descendant} & \text{genau dann, wenn } \text{inh}(v_2, v_1) = \text{Ancestor} \\ \text{Sibling} & \text{sonst} \end{cases} \quad (3.3)$$

Im Gegensatz zu allen anderen Constraints wird in diesem Fall kein numerisches Ähnlichkeitsmaß definiert, sondern eine strikte Bedingung gestellt: Zwei Knoten  $(v_{A1}, v_{A2})$  und  $(v_{B1}, v_{B2})$  des Assoziationsgraphen werden nicht verbunden, falls gilt:  $\text{inh}(v_{A1}, v_{B1}) \neq \text{inh}(v_{A2}, v_{B2})$ . Dabei ist es unerheblich, wie dieser Fall von anderen binären Constraints bewertet wird. Falls Gleichheit gilt, so wird mit der Analyse durch die anderen binären Constraints fortgefahren. Dem Inheritance – Constraint kommt somit eine Sonderrolle zu.

### 3.2.3. Bewertungsfunktion für Constraints

Wie in Abschnitt 2.3.2 auf Seite 17 definiert, sollen alle Constraints aus  $C_F$  und  $C_G$  als Wertebereich das Intervall  $[0, 1]$  besitzen. Viele Constraints berechnen intern das Verhältnis zweier Größen. Als Beispiel sei der EdgeDiameter – Constraint genannt, der den mittleren Durchmesser  $d_1$  und  $d_2$  der Kanten vergleicht, die in  $v_1$  bzw.  $v_2$  einmünden. Das Verhältnis  $d_1/d_2$  liegt im Intervall  $[0, \infty]$ .

Es wird nun eine Funktion  $a(r)$  angegeben, die aus dem Verhältnis  $r$  einer Eigenschaft von  $v_1$  und  $v_2$  den Wert  $f_i((v_1, v_2)) \in [0, 1]$  berechnet. In allen Fällen ist es so, dass  $r = 1$  ideal ist, da in diesem Fall beide Größen gleich sind. Demzufolge sollte  $a(1) = 1$  gelten. Weiterhin sollte  $a(r) = a(1/r)$  gelten, da es keinen Unterschied machen sollte, ob man  $v_1$  mit  $v_2$  in Verbindung setzt oder umgekehrt. Schließlich sollte gelten  $\lim_{r \rightarrow 0} a(r) = 0$  und  $\lim_{r \rightarrow \infty} a(r) = 0$ . Zur besseren Justierung der Werte von  $a(r)$  sei noch ein Parameter  $\sigma > 1$  gegeben, für den gelten soll:  $a(\sigma) = 0.5$ .  $\sigma$  ist also das größte Verhältnis, für das der Constraint das Matchen der Knoten  $v_1$  und  $v_2$  noch als sinnvoll bewertet. Auf diesem Weg lässt sich die Funktion  $a_\sigma(r)$  für jeden Constraint geeignet anpassen. Folgende Funktion erfüllt alle oben genannten Bedingungen, wie man leicht überprüft:

$$a_\sigma(r) = 2^{\frac{1 - \max(r, \frac{1}{r})}{\sigma - 1}} \quad (3.4)$$

### 3.2.4. Wahl eines Referenzknotens

Für verschiedene unäre Constraints ist es notwendig, einen Referenzknoten zu wählen. Ein solcher Referenzknoten sollte eine ausgezeichnete Rolle in dem jeweiligen Baum haben und eindeutig bestimmbar sein. Durch die Wahl eines Referenzknotens in beiden Bäumen wird implizit eine erste Zuordnung zweier Knoten etabliert, die das weitere Matching-Verfahren vereinfachen kann.

Eine nahe liegende Wahl für die Referenzknoten sind die Wurzeln der Bäume. Diese sind in den zugrunde liegenden Baumstrukturen bereits eindeutig bestimmt

und die Baumhierarchie somit definiert. Die Wurzeln der Bäume sind stets auf dem in das Organ einführenden bzw. herausführenden Gefäß platziert. Dieses Gefäß wird am "Rand" des Organs (oder einer vom Benutzer bestimmten Stelle) abgeschnitten und die Wurzel des Gefäßsystems kommt somit auf diesem Schnittpunkt zu liegen. Leider ist die Detektion des Randes während der Segmentierung nicht immer präzise. Daher ist die relative Position der Wurzel zu den anderen Knoten des Gefäßbaums nicht immer wohldefiniert. In vielen Fällen ist diese Ungenauigkeit vernachlässigbar gering, jedoch verbleibt eine Restunsicherheit.

Als Alternative bietet sich der *erste Hauptverzweigungsknoten* des Baumes an. Dies ist der Hauptverzweigungsknoten, der das geringste Level besitzt, also am nächsten zur Wurzel liegt. Ein Hauptverzweigungsknoten ist ein Knoten, an dem sich das Gefäßsystem in zwei (oder mehr) Teilsysteme aufspaltet, die alle eine signifikante Größe besitzen. Formal sei ein Hauptverzweigungsknoten wie folgt definiert:

Ein Knoten  $v$  ist ein Hauptverzweigungsknoten, wenn gilt:

$$size(T[v]) * c \leq \min_{v_i \text{ child of } v} size(T[v_i])$$

Hierbei sei  $T[v]$  der von  $v$  induzierte Teilbaum und  $size(T)$  die Summe der Längen der Kanten eines Baums  $T$ . Eine geeignete Wahl für  $c$  ist  $c \approx 0.2$ .

Die Wahl des ersten Hauptverzweigungsknoten ist – im Gegensatz zu den Hauptverzweigungsknoten zweiter und tieferer Stufe – stabil, sodass es in aller Regel sinnvoll ist, die Hauptverzweigungsknoten der beiden Bäume einander zuzuordnen. Ausnahmen treten lediglich beim Matching von prä- und postoperativen Gefäßsystemen auf, bei denen ein größerer Teil des Gefäßsystems entfernt wurde. In Abbildung 3.8 ist der erste Hauptverzweigungsknoten des Gefäßsystems rot eingefärbt. Man sieht, dass mehrere kleinere Abzweigungen ignoriert wurden und der erste Hauptverzweigungsknoten korrekt erkannt wurde.

### 3.3. Augmentierung der Clique

Nachdem eine maximale Clique im Assoziationsgraphen gefunden wurde – wie in Kapitel 4 erläutert wird – kann unmittelbar eine Zuordnung von Baumknoten abgeleitet werden. Mitunter ist es aber so, dass beim Erzeugen des Assoziationsgraphen die Constraints in manchen Teilbereichen der Bäume (wo die Unterschiede zwischen den beiden Bäumen vergleichsweise groß sind) zu restriktiv waren. Das führt dazu, dass maximale Cliques häufig klein sind und die abgeleiteten Knotenzuordnungen relativ große Bereiche des Baumes auslassen. Ein Lösungsansatz wäre es, die Constraints zu lockern. Das würde aber in anderen Bereichen des Baums die Qualität des Matchings verringern.

Stattdessen wird die ermittelte Clique in einem Nachbearbeitungsschritt noch um einige Knoten des Assoziationsgraphen ergänzt. Diese Knoten müssen folgende Eigenschaften besitzen:

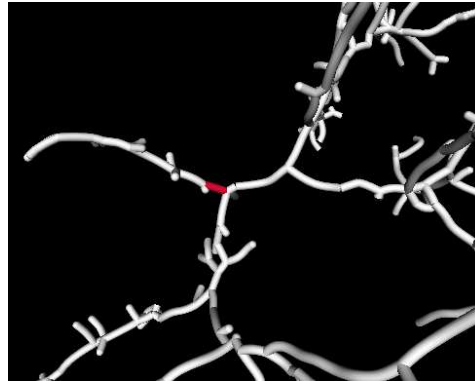


Abbildung 3.8.: Ein Hauptverzweigungsknoten

Die Wurzel des Gefäßsystems liegt in der Abbildung am linken Rand, der erste Hauptverzweigungsknoten des Gefäßsystems ist rot eingefärbt (Genauer: Die in den Hauptverzweigungsknoten einführende Kante ist rot eingefärbt). Mehrere kleinere Abzweigungen wurden ignoriert und der erste Hauptverzweigungsknoten korrekt erkannt.

1. Die beiden zu einem Assoziationsgraph-Knoten korrespondierenden Baumknoten sind jeweils noch nicht zugeordnet, das heißt, sie sind jeweils keinem der Knoten der ursprünglichen Clique zugeordnet.
2. Der Assoziationsgraph-Knoten besitzt einen hohen Verbindungsgrad zur ursprünglichen Clique, das heißt, er ist mit einem hohen Anteil  $p$  der Clique-Knoten verbunden.

Dieser Vorgang wird als *Augmentierung* (Anreicherung) bezeichnet.

Die aus dem Assoziationsgraph ausgewählte Knotenmenge ist nach einer Augmentierung natürlich keine Clique mehr. Ihre Knoten sind jedoch nichtsdestotrotz wechselseitig sehr stark verbunden, sodass das Matching weiterhin als konsistent angesehen werden kann.

Kritisch bei diesem Vorgehen ist die Wahl von  $p \in [0, 1]$ . Wird  $p$  zu groß gewählt, so wird die Clique kaum augmentiert. Wird  $p$  zu klein gewählt, so werden eventuell Knotenzuordnungen aufgenommen, die von den Constraints zu Recht ausgeschlossen wurden, und das Matching erscheint dadurch inkonsistent.

Da es sich als schwierig erwiesen hat, eine geeignete Wahl von  $p$  zu finden, die für alle Fälle sinnvoll ist (vergleiche Abschnitt 5.2.3), wurde die Augmentierung so in das Matching-Verfahren eingebaut, dass sie dynamisch vom Benutzer gesteuert werden kann. Dazu wird dem Benutzer zunächst das aus der Clique abgeleitete Matching-Resultat dargestellt<sup>16</sup>. Erscheint dem Benutzer dieses Ergebnis zu "spärlich", so kann er eine erste Augmentierung mit  $p = 0.95$  durchführen. Dies bedeutet, dass nur Knoten aufgenommen werden, die mit mindestens 95% der Knoten der Clique

<sup>16</sup>Zu den verschiedenen Visualisierungsmöglichkeiten eines Matchings siehe Abschnitt 5.1.

verbunden waren. Ist das Ergebnis weiterhin zu rar, so können weitere Augmentierungsschritte durchgeführt werden, wobei  $p$  sukzessive verringert wird. Somit kann der Benutzer dynamisch steuern, wann er mit dem abgeleiteten Matching zufrieden ist.





## Kapitel 4.

# Bestimmung einer maximalen Clique

In diesem Kapitel werden verschiedene Verfahren zur Detektion einer maximalen Clique vorgestellt und ihre Vor- und Nachteile dargelegt. Aus komplexitätstheoretischen Überlegungen weiß man, dass der Versuch, ein Verfahren zu finden, das in effizienter Weise das Problem *Maximum Clique* löst, also eine global maximale Clique<sup>1</sup> zu finden, zum Scheitern verurteilt ist (außer  $\mathcal{P} = \mathcal{NP}$ ). Aus diesem Grund werden verschiedene heuristische Verfahren vorgestellt, die im allgemeinen keine global maximale Clique finden, jedoch lokal maximale Cliques, deren Knoten-Kardinalität meist nur geringfügig unter der der global maximalen Cliques liegt.

### 4.1. Komplexitätstheoretische Überlegungen

Für das Problem *Clique* wurde 1972 als eines der ersten Probleme die  $\mathcal{NP}$ -Vollständigkeit gezeigt [Kar72]. Bei *Clique* handelt es sich um das *Entscheidungsproblem*, bei dem bestimmt werden soll, ob ein gegebener Graph  $G = (V, E)$  für einen Schwellwert  $c \in \mathbb{N}$  mit  $c \leq |V|$  eine Clique  $Q$  mit  $|Q| \geq c$  besitzt. Der Beweis wird durch eine einfache Reduktion  $3SAT \leq_P \text{Clique}$  geführt, für Details sei auf [Kar72] verwiesen.

Aus dieser Überlegung folgt unmittelbar, dass auch *Maximum Clique*, das Finden einer global maximalen Clique in einem Graphen,  $\mathcal{NP}$ -vollständig sein muss, da ansonsten ja die Kardinalität der größten Clique  $Q$  bekannt wäre und somit *Clique* in konstanter Zeit durch Vergleich von  $|Q|$  und  $c$  entscheidbar wäre.

Falls nicht  $\mathcal{P} = \mathcal{NP}$  gilt, existieren also keine Algorithmen mit polynomieller Laufzeit zur Bestimmung global maximaler Cliques. Da die Assoziationsgraphen, deren global maximale Cliques gefunden werden sollen, bis zu 50000 Knoten besitzen können, scheiden somit exakte Verfahren zur Lösung des Problems aus.

---

<sup>1</sup>Zur Definition einer global maximalen Clique siehe Appendix A auf Seite 70.

## 4.2. Kontinuierliche Charakterisierung des *Maximum Clique Problems*

Im Folgenden sollen verschiedene Heuristiken diskutiert werden, die auf das Problem *Maximum Clique* anwendbar sind. Eine Reihe heuristischer Ansatzpunkte wird zugänglich, wenn man das diskrete Problem *Maximum Clique* in ein äquivalentes kontinuierliches Problem umwandelt, wie in [Pel02] gezeigt. Einen Ansatzpunkt hierfür bietet das Motzkin-Straus Theorem [MS65]. Dieses erlaubt die Umwandlung des *Maximum Clique Problems* in ein kontinuierliches Problem der *quadratischen Programmierung*.

Sei  $G = (V, E)$  ein beliebiger Graph mit  $n = |V|$  und weiterhin  $\Delta$  der *Standard-Simplex* im  $\mathbb{R}^n$ :

$$\Delta = \{x \in \mathbb{R}^n : e'x = 1 \text{ und } x_i \geq 0, i = 1, \dots, n\} \quad (4.1)$$

wobei  $e = (1, \dots, 1)^T \in \mathbb{R}^n$  ist. Für eine nichtleere Teilmenge von Knoten  $C$  von  $V$ , bezeichne  $x^c \in \Delta$  den *charakteristischen Vektor* dieser Teilmenge, der wie folgt definiert ist:

$$x_i^c = \begin{cases} 1/|C| & \text{wenn } i \in C \\ 0 & \text{sonst} \end{cases} \quad (4.2)$$

Man betrachte nun die folgende quadratische Funktion

$$f_G(x) = x' A_G x + \frac{1}{2} x' x \quad (4.3)$$

wobei  $A_G$  die Adjazenzmatrix von  $G$  sei. Das folgende Theorem von Bomze [Bom97] erweitert das Motzkin-Strauss Theorem [MS65], und erlaubt einen eleganten Übergang vom diskreten Problem *Maximum Clique* auf ein kontinuierliches Problem.

**Theorem 1.** *Sei  $C \subseteq V$  eine Teilmenge von Knoten eines Graphen  $G = (V, E)$  und sei  $x^c$  ihr charakteristischer Vektor. Dann ist  $C$  eine global (lokal) maximale Clique von  $G$  genau dann, wenn  $x^c$  ein globales (lokales) Maximum von  $f_G$  auf  $\Delta$  ist. Weiterhin sind alle lokalen (und somit auch globalen) Maxima von  $f_G$  auf  $\Delta$  strikt und sind charakteristische Vektoren von maximalen Cliquen von  $G$ .*

Beweis: Siehe [Pel02].

Somit besteht also ein Eins-zu-Eins Zusammenhang zwischen maximalen Cliquen und Maxima von  $f_G$ . Der Übergang vom diskreten zum kontinuierlichen Problem stellt keine Näherung dar.

## 4.3. Heuristische Verfahren

Da keine effizienten Verfahren zur Bestimmung der global maximalen Clique bekannt sind, werden nachfolgend Verfahren vorgestellt, die mittels verschiedener Heuristiken

lokal maximale Cliques bestimmen, deren Kardinalität meist nur geringfügig unter der der global maximalen Clique liegt. Die verschiedenen Verfahren unterscheiden sich in Laufzeit und Güte der gefundenen Cliques.

### 4.3.1. Greedy Algorithmus

Ein erster nahe liegender Ansatz ist der Einsatz der *Greedy Heuristik*. Diese Heuristik garantiert für ein Problem genau dann das Finden eines globalen Optimums, wenn das Problem die Struktur eines *Matroids* besitzt. Dies ist im Fall von *Maximum Clique* jedoch nicht gegeben, da das Problem ansonsten nicht  $\mathcal{NP}$ -vollständig wäre.

Der Greedy-Algorithmus soll für einen Graphen  $G = (V, E)$  eine Teilmenge  $Q \subseteq V$  auswählen, die einerseits eine Clique ist und andererseits möglichst große Kardinalität besitzt. Benötigt wird eine *Bewertungsfunktion*  $y : V \rightarrow \mathbb{N}$ , die jedem Knoten eine "Güte" zuordnet. Der Greedy-Algorithmus wählt zu jedem Zeitpunkt das Element  $v \in V$ , für das  $y(v)$  möglichst groß ist - daher die Bezeichnung greedy (gierig).

Für  $y$  sind verschiedene Funktionen denkbar, nahe liegend für das Finden einer maximalen Clique ist es aber, jedem Knoten die Anzahl seiner Nachbarn zuzuordnen. Dieser Überlegung liegt zugrunde, dass Knoten, die viele Nachbarknoten besitzen, mit höherer Wahrscheinlichkeit auch in einer maximalen Clique enthalten sind. In Prozedur GreedyClique wird der Greedy-Algorithmus auf das Clique - Problem angewendet.

### 4.3.2. Evolutionäre Spieltheorie

Im Folgenden werden Methoden aus der *evolutionären Spieltheorie* verwendet, um eine quadratische Funktion  $f(x) = x'Ax$  - mit  $A$  symmetrisch - zu maximieren. Dies gibt in Zusammenhang mit Theorem 1 auf der vorherigen Seite eine effiziente Möglichkeit, eine maximale Clique zu bestimmen. Zunächst werden die grundlegenden Begriffe aus der evolutionären Spieltheorie kurz eingeführt. Der fundamentale Begriff der *Evolution* bezeichnet hierbei einen iterativen Optimierungsprozess. Die Darstellung orientiert sich an [Pel02].

Zu jedem Zeitpunkt  $t$  der Evolution sei eine *Population*  $P$  gegeben, die aus einer endlichen Menge an *Individuen* besteht. Jedes Individuum verfolgt eine bestimmte *Strategie*. Sei  $J = \{1, \dots, n\}$  die Menge möglicher Strategien und für alle  $i \in J$ , sei  $x_i(t)$  der Anteil an der Population  $P$ , der zu Zeitpunkt  $t$  die Strategie  $i$  spielt. Der Zustand einer Population zu einem gegebenen Zeitpunkt ist ein Vektor  $x = (x_1, \dots, x_n)'$ . Offensichtlich liegen alle Populationszustände  $x$  im Standardsimplex  $\Delta$ . Für einen gegebenen Populationszustand  $x \in \Delta$  definieren wir den *Support*  $\sigma$  von  $x$  als die Menge von nicht-ausgestorbenen Strategien:

$$\sigma(x) = \{i \in J : x_i > 0\}. \quad (4.4)$$

Sei  $A = (a_{ij})$  eine  $n \times n$  Payoff-Matrix. Das bedeutet, dass für jedes Paar  $i, j \in J$  von Strategien,  $a_{ij}$  den *Payoff* eines Individuums aus  $P$  bezeichnet, das Strategie

```

Input : Ein Graph  $G = (V, E)$ .
Output : Eine maximale Clique  $Q$ .
/* Berechne Bewertungsfunktion  $y$ , die jedem Knoten die Zahl seiner
   Nachbarn  $|N(i)|$  zuordnet */
Konstruiere Vektor  $y \in \mathbb{R}$  mit  $y_i = |N(i)|$ 
/* Initialisiere Menge der verfügbaren Knoten  $V_k$  und Clique  $Q = \emptyset$ 
   */
 $V_1 \leftarrow V$ ;
 $k \leftarrow 1$ ;
 $Q \leftarrow \emptyset$ ;
/* Solange noch Knoten zur Clique addierbar sind.. */
while  $V_k \neq \emptyset$  do
    Suche einen Knoten  $v_k \in V_k$ , sodass  $y_k$  maximal ist
     $Q \leftarrow Q \cup \{v_k\}$ ;
    /* Nur Knoten, die benachbart zu  $v_k$  sind, können in Zukunft zur
       Clique hinzugefügt werden */
     $V_{k+1} \leftarrow V_k \cap N(v_k)$ ;
    /* Aktualisiere  $y_j$  auf die Zahl der Nachbarn von  $v_j$ , die zur
       Clique addierbar sind */
    foreach  $v_j \in V_{k+1}$  do
         $y_j \leftarrow |N(j) \cap V_{k+1}|$ ;
     $k := k + 1$ 
return  $Q$ ;

```

#### Prozedur GreedyClique

$i$  gegen einen Konkurrenten mit Strategie  $j$  spielt. Wenn die Population in Zustand  $x$  ist, ist der erwartete Payoff bei einem Individuum mit Strategie  $i$ :

$$\pi_i(x) = \sum_{j=1}^n a_{ij}x_j = (Ax)_i \quad (4.5)$$

Der mittlere Payoff der gesamten Population ist

$$\pi(x) = \sum_{i=1}^n x_i \pi_i(x) = x' Ax \quad (4.6)$$

In der evolutionären Spieltheorie wird die Annahme gemacht, dass das Spiel immer und immer wieder gespielt wird, Generation für Generation, und dass die "natürliche Selektion" dafür sorgen wird, dass die beste Strategie überleben wird. Eine allgemeine Klasse von Gleichungen, die eine solche Evolution modellieren, wird durch die folgende Menge von gewöhnlichen Differentialgleichungen gegeben:

$$\dot{x}_i = x_i g_i(x), \quad (4.7)$$

wobei ein Punkt eine Ableitung nach der Zeit bedeutet und  $g = (g_1, \dots, g_n)$  eine Funktion auf  $\Delta$  ist. Anschaulich definiert  $g_i$  die Rate mit der die Strategie  $i$  sich repliziert, daher wird  $g_i$  im allgemeinen von  $\pi_i(x)$  abhängig sein. Gewöhnlich wird gefordert, dass  $g$  eine *reguläre* Funktion ist, was bedeutet, dass  $g$  Lipschitz-stetig ist und das  $g(x) * x = 0$  für alle  $x \in \Delta$  gilt. Die erste Bedingung garantiert, dass das System von Differentialgleichungen 4.7 eine eindeutige Lösung für jeden initialen Populationszustand besitzt. Die Bedingung  $g(x) * x = 0$  dagegen garantiert, dass der Standardsimplex unter 4.7 invariant ist, dass also jede Trajektorie, die in  $\Delta$  startet in  $\Delta$  verbleibt.

Gleichung 4.7 auf der vorherigen Seite wird als *payoff-monoton* bezeichnet, wenn gilt:

$$g_i(x) > g_j(x) \Leftrightarrow \pi_i(x) > \pi_j(x) \text{ für alle } x \in \Delta \quad (4.8)$$

Dies entspricht der vernünftigen Überlegung, dass Strategien mit höherem Payoff sich besser replizieren sollten. Es gilt folgendes Theorem:

**Theorem 2.** *Wenn die Payoff-Matrix  $A$  symmetrisch ist, dann ist  $\pi(x) = x'Ax$  strikt monoton wachsend entlang jeder nicht-konstanten Trajektorie einer payoff-monotonen Spieldynamik. In anderen Worten,  $\dot{\pi}(x(t)) \geq 0$  für alle  $t$ , mit Gleichheit genau dann, wenn  $x = x(t)$  ein stationärer Punkt ist.*

Beweis: Siehe [Pel02].

Eine bekannte Klasse von payoff-monotonen Spieldynamiken ist gegeben durch

$$\dot{x}_i = x_i \left( f(\pi_i(x)) - \sum_{j=1}^n x_j f(\pi_j(x)) \right), \quad (4.9)$$

wobei  $f(u)$  eine monoton wachsende Funktion in  $u$  ist. Wenn man  $f$  als die Identitätsfunktion setzt, erhält man die bekannten *Standard Replikator Gleichungen*

$$\dot{x}_i = x_i \left( \pi_i(x) - \sum_{j=1}^n x_j \pi_j(x) \right), \quad (4.10)$$

deren grundlegende Idee es ist, dass die durchschnittliche Wachstumsrate  $\frac{\dot{x}_i}{x_i}$  der Strategie  $i$  der Differenz zwischen durchschnittlichen Payoff der Strategie  $i$  und dem mittleren Payoff der gesamten Population entsprechen sollte.

Ein weiteres Populationsmodell erhält man, wenn man  $f(u) = e^{\kappa u}$  wählt; man erhält dann

$$\dot{x}_i = x_i \left( e^{\kappa \pi_i(x)} - \sum_{j=1}^n x_j e^{\kappa \pi_j(x)} \right), \quad (4.11)$$

wobei  $\kappa$  eine positive Konstante ist.

Zur Implementierung der payoff-monotonen Spieldynamiken werden folgende zeit-diskreten Modelle benutzt:

$$x_i(t+1) = \frac{x_i(t)\pi_i(t)}{\sum_{j=1}^n x_j(t)\pi_j(t)} \quad (4.12)$$

und

$$x_i(t+1) = \frac{x_i(t)e^{\kappa\pi_i(t)}}{\sum_{j=1}^n x_j(t)e^{\kappa\pi_j(t)}} \quad (4.13)$$

Diese Modelle korrespondieren zu bekannten Diskretisierungen von 4.10 und 4.11, für Näheres sei auf [HS98] verwiesen. Eine gute Wahl von  $\kappa$  ist  $\kappa = 10$ .

Ein geeigneter Startpunkt  $x(0)$  ist das *Baryzentrum* des Simplex, also  $x(0) = (\frac{1}{n}, \dots, \frac{1}{n})$ . Ein geeignetes Abbruchkriterium ist, wenn der  $\epsilon$ -Support

$$\sigma_\epsilon(x) = \{i \in J : x_i > \epsilon\}. \quad (4.14)$$

für kleines  $\epsilon$  sich über viele Iterationsschritte nicht mehr ändert.

Aufgrund des Theorems 2 auf der vorherigen Seite ist die evolutionäre Spieltheorie mit payoff-monotonen Spieldynamiken sehr gut als Heuristiken zum Lösen des *Maximum Clique Problems* geeignet. Setzt man  $A = A_G + \frac{1}{2}I$ , wobei  $I$  die Einheitsmatrix ist, so wissen wir aus Theorem 2, dass jede payoff-monotone Spieldynamik, startend von jedem beliebigen Startzustand  $x \in \Delta$ , iterativ die Funktion  $f_G$  aus Abschnitt 4.3 auf Seite 42 über dem Simplex maximiert und mit Sicherheit gegen ein striktes, lokales Maximum konvergiert, welches, aufgrund des Theorems 1, zu einem charakteristischen Vektor einer maximalen Clique von  $G$  korrespondiert.

**Anschauliche Interpretation** Im Folgenden soll kurz erläutert werden, warum obiges Vorgehen auch anschaulich sinnvoll ist. Betrachten wir dazu jeden Knoten des Assoziationsgraphen als Strategie. Im Laufe der Zeit kommt es zu wiederholten Interaktionen zwischen Individuen, die gewisse Strategien  $i$  und  $j$  spielen. Treffen zwei solche Individuen aufeinander, so ziehen sie daraus Nutzen (Payoff), wenn ihre Strategien kompatibel sind<sup>2</sup>. Ansonsten resultiert kein Nutzen. Je größer der Nutzen ist, den die Individuen mit Strategie  $i$  im Laufe der Zeit ansammeln, um so stärker steigt der Anteil dieser Strategie an der Population an<sup>3</sup>. Zum Anfang gebe es zu jeder Strategie genau gleich viele sie spielende Individuen. Zunächst werden sich offensichtlich Strategien, die mit möglichst vielen anderen kompatibel sind, ausbreiten. Strategien, die mit nur wenigen anderen Strategien kompatibel sind, werden dagegen aussterben. Im weiteren Verlauf werden dagegen auch Strategien, die ursprünglich mit vielen anderen Strategien kompatibel waren, deren "Partner" aber im Laufe der Zeit ausgestorben sind, ebenfalls aussterben. Auf Dauer wird sich also ein Zustand durchsetzen, in dem in der gesamten Population nur noch eine Gruppe von Strategien gespielt wird, die wechselseitig kompatibel sind. Dem entspricht aber genau eine Clique im Assoziationsgraph.

### 4.3.3. Genetische Algorithmen

Die kontinuierliche Charakterisierung des *Maximum Clique Problems* aus Abschnitt 4.2 legt nahe, *genetische Algorithmen* auf das Problem anzuwenden. Für eine aus-

<sup>2</sup>Zwei Strategien sind kompatibel, wenn die korrespondierenden Knoten  $i$  und  $j$  im Assoziationsgraphen verbunden sind.

<sup>3</sup>An dieser Stelle kommt die Payoff-Monotonität zum Tragen.

fürliche Einführung in das Gebiet der genetische Algorithmen sei auf [GKK04] verwiesen, hier soll nur kurz das Vorgehen für das *Maximum Clique* Problem zusammengefasst werden:

Als *Genom* wählen wir den charakteristischen Vektor der Clique gemäß Abschnitt 4.2 und als Fitnessfunktion  $f_G(x)$  aus Gleichung 4.3 auf Seite 42. Es gilt nun das Individuum zu finden, dessen Genom die Fitnessfunktion maximiert. Hierzu werden die Genome der Individuen der Anfangspopulation zufällig initialisiert. Danach werden in jedem Iterationsschritt die genetischen Operationen *Mutation* und *Crossover* auf die Individuen angewendet und am Ende eines Schrittes eine neue Population gebildet. In diese neue Population werden Individuen mit hoher Fitness wahrscheinlicher aufgenommen, als solche mit geringer Fitness. Nach Eintreten eines Abbruchkriteriums wird das Verfahren beendet und das Genom mit der höchsten Fitness repräsentiert den charakteristischen Vektor einer maximalen Clique.

Parameter für das Verfahren sind die Größe der Population sowie die Mutations- und Crossover-Wahrscheinlichkeit.

## 4.4. Vergleich der verschiedenen Verfahren

Im Rahmen dieser Diplomarbeit wurden die drei oben erläuterten heuristischen Ansätze zur Detektion einer maximalen Clique implementiert. Ausgehend von zwei Paaren von Leber-Datensätzen wurden durch das in Kapitel 3 beschriebene Verfahren zwei Assoziationsgraphen erzeugt. Die drei Verfahren wurden an diesen beiden Graphen hinsichtlich Laufzeit und Güte des Ergebnisses verglichen.

Als Maß für die Güte der gelieferten Ergebnisse bietet sich intuitiv die Größe der gefundenen Knotenmenge – also die Anzahl gematchter Knoten – an. Die vom Greedy-Verfahren gelieferte Knotenmenge ist garantiert eine Clique; die beiden anderen Ansätze sind iterative Verfahren, die gegen eine Clique konvergieren. Als kritisch hat sich bei diesen die Wahl des Abbruchkriteriums erwiesen: Bricht man die Verfahren zu früh ab, so ist die gelieferte Knotenmenge meist keine Clique, sondern nur ein Teilgraph der sehr stark verknüpft ist. Daher sollte neben der Kardinalität der Knotenmenge auch ihr Verbindungsgrad berücksichtigt werden. Ein solches Maß ist genau die Funktion  $f_G(x)$  aus Gleichung 4.3 auf Seite 42. Je größer  $f_G(x)$  ist, desto besser ist der Verbindungsgrad der Knotenmenge.

Für das Verfahren aus der Evolutionären Spieltheorie wurde als Abbruchkriterium gewählt, dass die Population über einen Zeitraum von 1000 Generationen stabil ist, also keine Strategie mehr ausgestorben ist. Des Weiteren werden als Spieldynamik die Standard Replikator Gleichungen verwendet. Für die Genetischen Algorithmen wurde als Abbruchkriterium gewählt, dass über einen Zeitraum von 500 Iterationen kein neues Genom mehr aufgetreten ist, dass eine höhere Fitness besitzt, als alle vorherigen. Weiterhin wurde eine Populationsgröße von 150 Individuen gewählt, eine Crossover-Wahrscheinlichkeit von 0.5 sowie eine Mutations-Wahrscheinlichkeit von 0.2.

Aus den Tabellen 4.1 und 4.2 lassen sich folgende Informationen ablesen:

	Laufzeit	Größe der Knotenmenge	$f_g$
Greedy	2,62 sec	50	0,99
Evolutionäre Spieltheorie	5,08 sec	70	0,9929
Genetische Algorithmen	587,15 sec	49	0,9756

Tabelle 4.1.: Laufzeit und Güte der Ergebnisse der Clique-Algorithmen für einen Assoziationsgraph mit 3502 Knoten

	Laufzeit	Größe der Knotenmenge	$f_g$
Greedy	36,93 sec	83	0,994
Evolutionäre Spieltheorie	131,06 sec	114	0,9956
Genetische Algorithmen	» 5000 sec	91	0,9588

Tabelle 4.2.: Laufzeit und Güte der Ergebnisse der Clique-Algorithmen für einen Assoziationsgraph mit 14268 Knoten

- Die Ergebnisse des Genetische Algorithmen - Verfahrens sind schlechter als die der anderen Verfahren obwohl die Laufzeit – insbesondere für große Eingabegraphen – unverhältnismäßig hoch ist.
- Das Greedy – Verfahren benötigt die geringste Laufzeit, allerdings sind die gelieferten Ergebnisse deutlich schlechter als die des Evolutionäre Spieltheorie - Verfahrens.
- Das Verfahren aus der Evolutionären Spieltheorie liefert die besten Ergebnisse; zwar ist das Ergebnis nicht garantiert eine Clique, jedoch sind die Ergebnisse hinreichend ähnlich zu einer solchen. So ist es meist ausreichend einige wenige Knoten aus der Knotenmenge zu entfernen, damit es sich um eine Clique handelt. Diese Nachbearbeitung ist in den Tabellen bereits berücksichtigt.

Aufgrund dieser Ergebnisse empfiehlt es sich, wann immer möglich das Verfahren aus der Evolutionären Spieltheorie zu benutzen. Lediglich, wenn der Nutzen einer geringen Laufzeit deutlich höher ist als der einer hohen Qualität des Ergebnisses kann das Greedy-Verfahren eine Alternative darstellen. Dieses Ergebnis wird auch durch den Vergleich mit einem Ground-Truth in Abschnitt 5.2.3 bestätigt.



## Kapitel 5.

# Anwendung auf Gefäßsysteme in Leber und Lunge

In diesem Kapitel sollen die Ergebnisse des Matching-Verfahrens bewertet und optimiert werden. Dazu wurden zunächst zwei Verfahren realisiert, welche die Ergebnisse des Matching-Verfahrens visualisieren, um so dem Benutzer eine Vorstellung von der Güte des Matchings zu ermöglichen. Diese Visualisierungen basieren auf einer Einfärbung der Kanten eines Baumes und werden in Abschnitt 5.1 vorgestellt werden.

Als Grundlage für die empirische Optimierung der Constraints aus Kapitel 3 lag für zwei Baumpaare ein manuell erstelltes Matching als *Ground Truth* vor. Diese Optimierung sowie eine Bewertung der Güte des Matchings im Vergleich zum Ground-Truth wird in Abschnitt 5.2.2 vorgenommen.

### 5.1. Visualisierungen

In diesem Abschnitt werden zwei Ansätze vorgestellt, mittels derer ein Matching visualisiert werden kann. In *MeVisLab* existieren bereits Module, die ein Gefäßsystem darstellen können und gewisse Interaktionen mit diesem erlauben. Daher war es nahe liegend, diese Module zur Visualisierung zu nutzen. Die Färbung einer Kante in der Visualisierung lässt sich über ein Attribut in der Graphstruktur steuern.

#### 5.1.1. Darstellung des Matchings

Ein Matching besteht aus einer Menge von Knotenzuordnungen. Jede Zuordnung zweier Knoten impliziert jedoch automatisch auch eine Zuordnung der in die Knoten eingehenden Kanten. Ein erster Ansatz ist es, jeder Knotenzuordnung eine eindeutige Farbe zuzuordnen und mit dieser Farbe die eingehende Kante einzufärben<sup>1</sup>. Dieser Ansatz hat allerdings verschiedene Nachteile, die jedoch leicht behoben werden können:

- Aufgrund der großen Anzahl von zugeordneten Knoten würde eine hohe Anzahl verschiedener Farben benötigt, um jeder Kante eine eindeutige Farbe

---

<sup>1</sup>Dieses Vorgehen bedingt natürlich, dass eine Zuordnung eines Wurzelknotens mangels eingehender Kante nicht repräsentiert ist.

zuzuordnen. Das hätte zur Folge, das ähnliche Farben vom menschlichen Betrachter nicht mehr eindeutig unterschieden werden könnten. Daher wird nur eine feste Menge von Farben verwendet, die für mehrere Zuordnungen verwendet werden. Dadurch ist die Zuordnung gleich gefärbter Kanten nicht mehr eindeutig, jedoch ist es für den Betrachter dennoch meist möglich, die korrespondierenden Kanten zu identifizieren. Um verbleibende Zweideutigkeiten zu beseitigen, besteht im interaktiven Modus in MeVisLab die Möglichkeit, eine Kante anzuklicken. Ist ihr Endknoten im Matching einem Knoten des anderen Baums zugeordnet, so werden diese beiden Knoten jeweils durch einen Marker hervorgehoben.

- Eingehende Kanten von zugeordneten Knoten besitzen mitunter deutlich verschiedene Längen; um die Ähnlichkeit zu betonen ist es sinnvoll, nur gleich lange Bereiche der Kanten einzufärben.
- Eine Einfärbung nur auf der Ebene zugeordneter Kanten ist – aufgrund der hohen Anzahl an Kanten – relativ unübersichtlich. Daher ist es wünschenswert, dass Teilbereiche der Bäume, die zueinander topologisch isomorph<sup>2</sup> sind und die vom Matching als solche erkannt worden sind, in einer gemeinsamen Farbe gefärbt werden. Dies erleichtert es dem Betrachter, Ähnlichkeiten der beiden Bäume zu erkennen.

Beispiele für auf diese Weise visualisierte Matching-Ergebnisse sind in Abbildung 5.2 und 5.3 dargestellt.

### 5.1.2. Übertragung einer Baumfärbung

Ein weiterer Ansatz ist die Übertragung einer bestehenden Färbung des einen Baums auf den anderen. Viele Gefäßbäume sind bereits eingefärbt; beispielsweise wurde für viele Leber-Gefäßbäume eine Einfärbung der Lebersegmente nach Couinaud/Bismuth durchgeführt (vergleiche [Sel00]). Die Übertragung einer solchen Färbung ist insbesondere dann sinnvoll, wenn eine automatisierte Einfärbung der Lebersegmente schwierig ist. Da eine manuelle Einfärbung aufwändig ist, ist es wünschenswert, dies nur für einen der Bäume durchzuführen und diese Färbung mittels des Matchings auf den/die anderen Bäume zu übertragen. Hierzu wird jeder Kante  $e$  des zu färbenden Baums auf folgende Weise eine Farbe zugeordnet:

Seien zwei Kanten wieder genau dann einander zugeordnet, wenn ihre Endknoten gematcht sind. Ist der Kante  $e$  eine Kante im gefärbten Baum zugeordnet, so übernehme deren Farbe für  $e$ . Anderenfalls wird die erste gematchte Kante  $e_a$  auf dem Pfad von  $e$  zur Wurzel gesucht. Dann wird die Farbe der  $e_a$  zugeordneten Kante festgestellt und auf  $e$  übertragen. Falls keine solcher Kante  $e_a$  existiert, so verbleibt die Kante ungefärbt. Dieses Vorgehen lässt sich wie folgt zusammenfassen: Übertrage auf jede gematchte Kante die Farbe der zugeordneten Kante und lasse dann im zu färbenden Baum die Färbungen durch den Baum absinken.

<sup>2</sup>Zum Begriff der topologischen Isomorphie sei auf Seite 69 verwiesen.

Bei dieser Visualisierung ist natürlich keine Zuordnung der einzelnen Knoten mehr erkennbar, dafür wird aber deutlich, welche Regionen (Lebersegmente) einander zugeordnet wurden. Um möglichst große Bereiche des Baums zu färben, ist es meist sinnvoll, das Matching noch wie in Abschnitt 3.3 beschrieben zu augmentieren. Das Ergebnis des Transfers einer Einfärbung der Lebersegmente nach Couinaud/Bismuth von einem Portalvenen-Baum auf einen anderen ist in Abbildung 5.1 dargestellt. Es ist nur noch eine geringe manuelle Überarbeitung erforderlich.

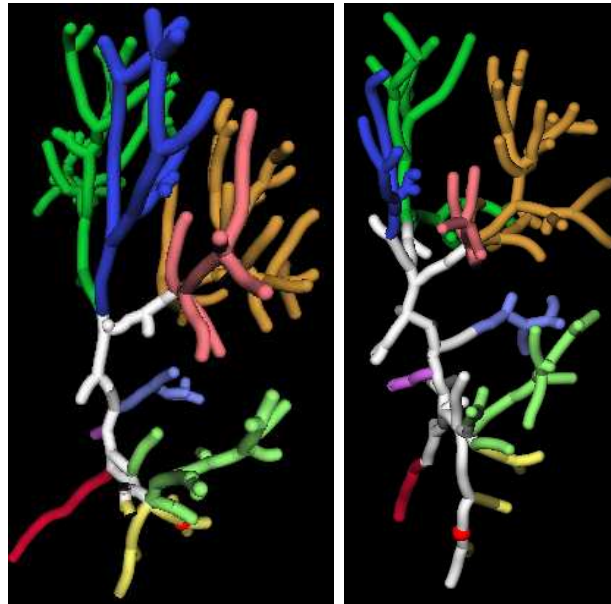


Abbildung 5.1.: Transfer einer Einfärbung der Lebersegmente nach Couinaud/-Bismuth.

Links die der Portalvenenbaum dargestellt, der als Quelle der Färbung fungiert. Rechts ist die Übertragung der Einfärbung auf den anderen Portalvenenbaum abgebildet.

## 5.2. Evaluierung

In diesem Abschnitt sollen Aussagen über die Eignung des Assoziationsgraph-Verfahrens für die konkrete Aufgabenstellung getroffen werden. Hierzu wird in Abschnitt 5.2.2 zunächst untersucht, welche Constraints für die Problemstellung geeignet sind, und wie gewisse Parameter zu wählen sind. Anschließend werden in 5.2.3 mit der ermittelten Sammlung von Constraints andere Konfigurationsmöglichkeiten, wie zum Beispiel die Wahl des Referenzknotens oder die Verfahren zum Finden einer maximalen Clique, verglichen. In der Regel werden nur die Ergebnisse des Leber-Portalvenen-Baums aufgeführt, da die Ergebnisse des Bronchialbaums ähnlich waren. Unterschiede zwischen diesen beiden Fällen werden explizit erwähnt.

### 5.2.1. Ground Truth

Als Ground Truth lagen zwei per Hand erstellte Matchings vor, je eins für ein Paar von Bronchialbäumen<sup>3</sup> und eins für ein Paar von Portalvenen-Bäume<sup>4</sup>. Diese bestehen jeweils aus 34 Zuordnungen, wobei beide Bäume circa 200 Knoten besitzen. Diese Zuordnungen wurden primär auf Basis der Gefäßbäume gewählt; nur in Fällen, wo auch dem menschlichen Betrachter die Zuordnung auf Basis der Gefäßsysteme zweideutig erschien, wurden die ursprünglichen CT/MRT-Bilddaten zu Rate gezogen. Zugeordnet wurden in erster Linie innere Verzweigungsknoten der Graphen, da gerade auf den Blättern nicht notwendigerweise eine identische Darstellungsgenauigkeit für beide Graphen gegeben ist.

### 5.2.2. Optimierung der Constraints mittels Ground-Truth Daten

In diesem Unterabschnitt wird – basierend auf dem Ground-Truth – die Güte der einzelnen Constraints analysiert. Aufgrund des großen Parameterraums<sup>5</sup> ist die Suche nach einem global optimalen Parametersatz – bezüglich des Ground Truth – sehr zeitaufwändig.

Stattdessen wird auf folgendem Weg ein "nahezu optimaler" Parametersatz bestimmt: Zunächst werden die unären und binären Constraints isoliert betrachtet, ein optimaler Parameter  $\sigma$  bestimmt und die Eignung des jeweiligen Constraints zur Bestimmung eines adäquaten Matchings untersucht. Hierauf aufbauend werden verschiedene Kombinationen von Constraints ausgewählt und die Güte des resultierenden Matchings bewertet.

#### Analyse der unären Constraints

Die in Kapitel 3.2.1 vorgestellten unären Constraints dienen der Identifizierung von Paaren von Baumknoten, die potentiell in einem sinnvollen Matching enthalten sein könnten. Den unären Constraints kommt daher eine entscheidende Rolle zu: Werden sinnvolle Knotenzuordnungen fälschlicherweise ausgeschlossen, so sind sie nicht im Assoziationsgraph repräsentiert und können somit auch im ermittelten Matching nicht enthalten sein. Werden dagegen zu viele Knotenpaare akzeptiert, so steigt die Größe des Assoziationsgraphen und somit die Laufzeit. Zudem ist es im weiteren Verlauf des Verfahrens schwieriger, ein gutes Matching aus dem Assoziationsgraph abzuleiten.

Daher wird im folgenden jeder unäre Constraints isoliert untersucht. Das Matching-Verfahren wird, wie in Kapitel 3 beschrieben, für ein Paar von Leber-Portal-

---

<sup>3</sup>Die beiden Bronchialbäumen stammen aus einer Aufnahme der Lunge im eingeatmeten Zustand und einer Aufnahme des ausgeatmeten Zustands mittels desselben bildgebenden Verfahrens.

<sup>4</sup>Die beiden Portalvenenbäumen stammen aus einer Aufnahme der Leber mittels Magnetresonanztomographie (MRT) und einer Aufnahme mittels Computertomographie (CT).

<sup>5</sup>Parameter sind die  $\sigma$  der einzelnen Constraints sowie die relative Gewichtung (Stimmenzahl) der Constraints.

venenbäumen durchgeführt; allerdings wird statt einer Kombination unärer Constraints jeweils nur ein einzelner berücksichtigt. Das erlaubt es, eine grobe Bewertung der Güte dieses unären Constraints zu erhalten. Das resultierende Matching wird mit dem aus 34 Zuordnungen bestehenden Ground-Truth verglichen. Relevante Ergebnisse sind die Anzahl korrekter und falscher Zuordnungen sowie die Laufzeit, die die Überprüfung aller Knotenpaare durch den jeweiligen Constraint benötigt. Des weiteren ist in Tabelle 5.1 der Parameter  $\sigma$  der Bewertungsfunktion aus Abschnitt 3.2.3 aufgeführt, für den die Ergebnisse des jeweiligen Constraints optimal waren.

Constraint	$\sigma$	Korrekt	Fehler	Laufzeit
EdgeDiameter	1.5	0	3	0.08 sec
BifurcationAngle	1.5	0	5	0.18 sec
SubtreeSimilarity	1.1	4	2	2.36 sec
BifurcationSimilarity	1.2			
RN-PathCurvature	1.08	7	3	0.16 sec
RN-CurveSimilarity	1.04	11	1	14.79 sec
RN-CurveTangent	1.15	18	4	2.58 sec

Tabelle 5.1.: Vergleich unärer Constraints

Der EdgeDiameter-Constraint führt in diesem Fall offenbar zu keinen brauchbaren Ergebnissen. Der Grund hierfür dürfte sein, dass Äste der segmentierten Gefäßbäume (im Falle der dem Ground-Truth zugrunde liegenden Bäume) auch in korrespondierenden Bereichen deutlich verschiedene Durchmesser besitzen (siehe auch Abbildung 5.2). Die Ursache liegt in den verschiedenen bildgebenden Verfahren, mittels derer das zugrunde liegende Bildmaterial aufgenommen wurde<sup>6</sup>. Ähnlich schlechte Ergebnisse liefert auch der BifurcationAngle-Constraint. Hierbei scheint die Ursache zu sein, dass viele Verzweigungspunkte sehr ähnliche Winkel besitzen und diese Winkel auch keine invariante Eigenschaft sind. Lokale Eigenschaften sind demnach vergleichsweise ungeeignet.

Etwas bessere Ergebnisse liefern die Constraints, die auf der Größe von Teilbäumen basieren. Deren Ergebnisse sind in Tabelle 5.1 nicht isoliert aufgeführt, sondern gemeinsam. Der Grund hierfür ist, dass diese Constraints isoliert relativ schlechte Ergebnisse liefern, kombiniert aber deutlich bessere. Die aufgeführten Ergebnisse resultieren aus einer Gewichtung von 1:1 der beiden Constraints, die das beste Resultat lieferte.

Die besten Ergebnisse resultieren aus den Constraints, die die relative Lage der Knoten bezüglich eines Referenzknotens vergleichen. Der RN-PathCurvature-Constraint besitzt wie erwartet die deutlich geringste Laufzeit, jedoch liefert er auch schlechtere Ergebnisse als die beiden anderen Constraints, die auf einem Referenz-

<sup>6</sup>CT und MRT liefern bei bestimmten Portalvenenbäume deutlich verschiedene Durchmesser. Dies ist bei diesem speziellen Datensatz der Fall, im allgemeinen sollten die Durchmesser ein besseres Kriterium darstellen.

knoten beasieren. Diese liefern vergleichbar gute Ergebnisse, jedoch ist die Laufzeit des RN-CurveTangent-Constraint deutlich geringer. Die Ursache hierfür ist, dass bei diesem die Optimierung bezüglich des Parameters  $\phi$  entfällt.

Im nächsten Schritt soll nun untersucht werden, inwiefern die Kombination verschiedener Constraints zu Verbesserungen führen kann. Es hat sich gezeigt, dass nur die drei Constraints, die sich auf einen Referenzknoten beziehen, zu sinnvollen Kombinationen beitragen. Daher werden hier exemplarisch die vier folgenden Kombinationen dieser Constraints untersucht. Hierbei ist in Klammern die relative Gewichtung (die  $\omega_i$  aus Definition 2 auf Seite 17) der Constraints aufgeführt. Die jeweiligen  $\sigma$  wurden gewählt wie in Tabelle 5.1.

*Konfiguration A1:* RN-PathCurvature (1) : RN-CurveTangent (1)

*Konfiguration B1:* RN-PathCurvature (1) : RN-CurveTangent (4)

*Konfiguration C1:* RN-PathCurvature (1) : RN-CurveSimilarity (4)

*Konfiguration D1:* RN-CurveSimilarity (1) : RN-CurveTangent(1)

Konfiguration	Korrekt	Fehler
A1	13	2
B1	20	2
C1	10	4
D1	16	4

Tabelle 5.2.: Kombination unärer Constraints

In Tabelle 5.2 sind die Ergebnisse aufgeführt. Die besten Ergebnisse liefert Kombination B1; nur in dieser ist eine Verbesserung gegenüber der isolierten Betrachtung der Constraints gegeben.

### Analyse der binären Constraints

Analog zu der Untersuchung der unären Constraints im vorhergehenden Abschnitt werden in diesem Abschnitt zuerst die binären Constraints isoliert bewertet<sup>7</sup>. Hierbei wird als Grundlage wiederum der Ground-Truth der Leberdaten benutzt und als unäre Constraints die Konfiguration B1 von oben eingesetzt. Die Ergebnisse sind in Tabelle 5.3 zusammengefasst.

PathCurvature- und CurveTangent-Constraint liefern beide gute Ergebnisse, allerdings ist die Laufzeit des letzteren – wie zu erwarten war – vergleichsweise hoch. Der PathLength – Constraint ist isoliert kein sinnvolles Maß; ist aber in Kombination mit den anderen Constraints durchaus sinnvoll. Im Folgenden werden Kombinationen binärer Constraints untersucht. Folgende Konfigurationen wurden exemplarisch gewählt:

<sup>7</sup>Der Inheritance-Constraint ist von dieser Untersuchung ausgenommen, da ihm eine Sonderrolle zukommt. Er ist standardmäßig bei allen folgenden Messungen aktiviert.

Constraint	$\sigma$	Korrekt	Fehler	Laufzeit
PathLength	1.7	2	4	6 sec
PathCurvature	1.3	18	2	8.9 sec
CurveTangent	1.13	17	2	203.97 sec

Tabelle 5.3.: Vergleich binärer Constraints

*Konfiguration A2:* PathCurvature (1) : CurveTangent (1)

*Konfiguration B2:* PathCurvature (3) : PathLength (1)

*Konfiguration C2:* CurveTangent (3) : PathLength (1)

Konfiguration	Korrekt	Fehler
A2	17	4
B2	20	2
C2	17	0

Tabelle 5.4.: Kombination binärer Constraints

In Tabelle 5.4 sind die Ergebnisse der verschiedenen Konfigurationen zusammengefasst. Die besten Ergebnisse liefert Konfiguration C2, allerdings ist diese auch mit einer deutlich höheren Laufzeit des Verfahrens verbunden, weil von dem zeitaufwändigen CurveTangent – Constraint Gebrauch gemacht wird. Alternativ kommt auch Konfiguration B2 zu brauchbaren Ergebnissen, jedoch mit einer deutlich geringeren Laufzeit.

### 5.2.3. Analyse verschiedener Konfigurationsmöglichkeiten

In diesem Abschnitt werden verschiedene Konfigurationsmöglichkeiten untersucht. Als unäre Constraints wird dabei stets die Konfiguration B1 und als binäre Constraints die Konfiguration C2 verwendet. Aufgrund der Verwendung der Konfiguration C2 sind die Laufzeiten relativ hoch.

#### Pruning

Das in Abschnitt 3.1 vorgestellte Pruning dient dazu, Gefäßäste, die aus Segmentierungsfehlern resultieren, zu entfernen und somit die Problemgröße zu reduzieren. Wie in Abschnitt 3.1 erläutert, führt dies eher bei Bronchialbäumen zu Verbesserungen, weniger bei Gefäßsystemen der Leber. Dies wird auch von den Ergebnissen in Tabelle 5.5 bestätigt. In jedem Falle verringert sich jedoch die Laufzeit deutlich.

Pruning Portalvene	Korrekt	Fehler	Laufzeit
ja	15	0	119.5 sec
nein	17	0	207.92 sec

Pruning Bronchialbaum	Korrekt	Fehler	Laufzeit
ja	16	3	371.14 sec
nein	14	3	936.52 sec

Tabelle 5.5.: Nutzen des Prunings

Die obere Tabelle zeigt die Ergebnisse des Matching-Verfahrens bei aktiviertem bzw. deaktiviertem Pruning für eine Portalvene. Die untere Tabelle zeigt dasselbe für den Fall eines Bronchialbaums. Bei der Portalvene kommt es zu einer Verschlechterung der Ergebnisse; beim Bronchialbaum werden die Ergebnisse deutlich besser. Die Laufzeit sinkt in beiden Fällen deutlich.

### Wahl des Referenzknotens

In Abschnitt 3.2.4 wurden zwei verschiedene Optionen für den Referenzknoten vorgestellt: Zum einen die Wurzel und zum anderen der erste Hauptverzweigungsknoten des Baums. Entgegen den Erwartungen hat sich sowohl beim Portalvenen-Baum als auch beim Bronchialbaum die Wahl der Wurzel als Referenzknoten als vorteilhaft erwiesen (vgl. Tabelle 5.6). Eine mögliche Erklärung hierfür ist, dass bei beiden Datensätzen die Wurzel gut gewählt wurde und die eingehenden Gefäße gleich lang sind. Bei anderen Datensätzen, bei denen dies nicht der Fall ist, könnte die Wahl des ersten Hauptverzweigungsknoten besser sein.

Referenzknoten	Korrekt	Fehler
Wurzel	17	0
Hauptverzweigungsknoten	13	0

Tabelle 5.6.: Wahl des Referenzknotens

### Verfahren zur Detektion einer maximalen Clique

Die Ergebnisse des Vergleichs der verschiedenen Verfahren zur Detektion einer maximalen Clique aus dem Abschnitt 4.4 werden auch durch den Vergleich des resultierenden Matchings mit dem Ground-Truth in Tabelle 5.7 bestätigt. Den besten Kompromiss zwischen Laufzeit und Güte des Ergebnisses stellt das Verfahren aus der Evolutionären Spieltheorie dar.



Verfahren	Korrekt	Fehler	Laufzeit
Greedy	13	2	0.33 sec
Evolutionäre Spieltheorie	17	0	0.59 sec
Genetische Algorithmen	9	2	173.89 sec

Tabelle 5.7.: Vergleich der Verfahren zur Detektion einer maximalen Clique

### Augmentierung

Die in Abschnitt 3.3 vorgestellte Augmentierung eines Matchings dient dazu, die Anzahl gematchter Knoten zu erhöhen. Dies sollte möglichst nur zu einer Erhöhung der korrekt gematchten Knoten führen und die Anzahl der Fehler nicht vergrößern. In Tabelle 5.8 sind diese beiden Größen in Abhängigkeit von der Zahl der Augmentierungsschritte eingetragen. Bis zu zwei Augmentierungsschritte sind im Fall des Bronchialbaums sinnvoll, da fünf zusätzlichen korrekten Zuordnungen nur ein zusätzlicher Fehler gegenübersteht. Im Falle der Portalvenen-Bäume war eine Augmentierung jedoch nicht erfolgreich.

Augmentierungen	Korrekt	Fehler	Augmentierungen	Korrekt	Fehler
0	16	3	0	17	0
1	20	4	1	17	0
2	21	4	2	17	1
4	21	6	4	17	3
7	21	10	7	19	3

Tabelle 5.8.: Analyse des Augmentierungsverfahrens

Links ist die Augmentierung des Bronchialbaum-Matchings zusammengefasst, rechts die des Matchings der beiden Portalvenen-Bäume.

## 5.3. Ergebnisse

In Abschnitt 5.2 wurden geeignete Wahlen von Constraints und Einstellungen bestimmt. Die Ergebnisse des Matching-Verfahrens unter diesen Einstellungen sollen nun am Beispiel des Portalvenen-Baums und des Bronchialbaums zusammengefasst und visualisiert werden.

Die Ergebnisse des Matching-Verfahrens sind Tabelle 5.9 zusammengefasst. Die Qualität des Ergebnisses ist im Falle der Portalvenen-Bäume besser als im Falle der Bronchialbäume. Ob dies nur in diesen beiden Fällen oder auch allgemein so ist, müsste mit Ground-Truth Daten weiterer Bäume überprüft werden. Eine mögliche Erklärung ist die Dichotomie der Bronchialbäume, die dazu führt, dass sich viele

Bereiche innerhalb eines Baumes sehr ähneln<sup>8</sup>. Dies erschwert das Matching.

Datensatz	Korrekt	Fehler	Ausgelassen	$\otimes$ top. Distanz	Laufzeit
Portalvene	17	0	17	–	202.35 sec
Bronchialbaum	21	4	9	1.25	369.23 sec

Tabelle 5.9.: Ergebnisse des Matchingverfahrens

Nichtsdestotrotz sind die Ergebnisse auch im Falle des Bronchialbaums gut. Zwar sind von den 25 Zuordnungen 4 falsch, jedoch sind diese Fehler nur geringfügig. Um dies zu begründen, betrachten wir diese Fehlzuordnungen genauer: Das Matching-Verfahren liefere die Zuordnung  $v \leftrightarrow v_1$ , im Ground-Truth sei stattdessen die Zuordnung  $v \leftrightarrow v_2$  enthalten. Dann liefert die topologische Distanz  $d_t(v_1, v_2)$  eine Maß für die Schwere des Fehlers. Die durchschnittliche topologische Distanz der falsch zugeordneten Knoten im Bronchialbaum beträgt 1.25 (vergleiche Tabelle 5.9). Das bedeutet, dass die Zuordnung meist nur um eine Ebene nach oben oder unten in der Baumhierarchie verrutscht ist. Meist ist der Pfad zwischen  $v_1$  und  $v_2$  zudem relativ kurz, d. h.  $v_1$  und  $v_2$  liegen in unmittelbarer Nähe, was eine Vertauschung weniger schwerwiegend macht.

In Abbildung 5.2 ist das Matching der Portalvenen-Bäume mittels des in Abschnitt 5.1.1 vorgestellten Verfahrens visualisiert. In Abbildung 5.3 ist auf die gleiche Weise das Matching der Bronchialbäume dargestellt.

## 5.4. Bewertung

Die Ergebnisse dieses Kapitels zeigen, dass das implementierte Verfahren für typische Beispiele von Gefäßbäumen aus Leber und Lunge in der Lage ist, die Anforderungen aus Abschnitt 1.4 zu erfüllen: Es wird ein signifikanter Anteil der Baumknoten mit einem akzeptablem Zeitaufwand sinnvoll zugeordnet. Das Verfahren liefert dabei ein Matching, das keine bzw. nur wenige geringe Fehlzuordnungen beinhaltet und die meisten Teilbereiche der Bäume abdeckt.

Zu untersuchen bleibt, inwiefern mit diesem Verfahren ähnliche Ergebnisse auch für schwierigere Anwendungsfälle zu erreichen sind. Als Beispiele seien hierfür die Aufnahme eines Organs mit verschiedenen bildgebenden Modalitäten mit deutlich verschiedenen Auflösungen<sup>9</sup> sowie das Matching von Regenerationsdaten der Leber genannt. Für eine systematische Untersuchung solcher Eingabedaten wäre ein manuell erstelltes Matching als Ground Truth notwendig; die Visualisierungen der Ergebnisse des Matching-Verfahrens sehen recht viel versprechend aus. Allerdings sind hierfür Modifikationen an einigen Parametern erforderlich.

<sup>8</sup>Ein Beispiel für ähnliche Teilbäume sind die rechte und die linke Hälfte des Bronchialbaum, die sich am ersten Hauptverzweigungspunkt trennen.

<sup>9</sup>Verschiedene Auflösungen implizieren eine deutlich unterschiedliche Anzahl Knoten in den Bäumen.

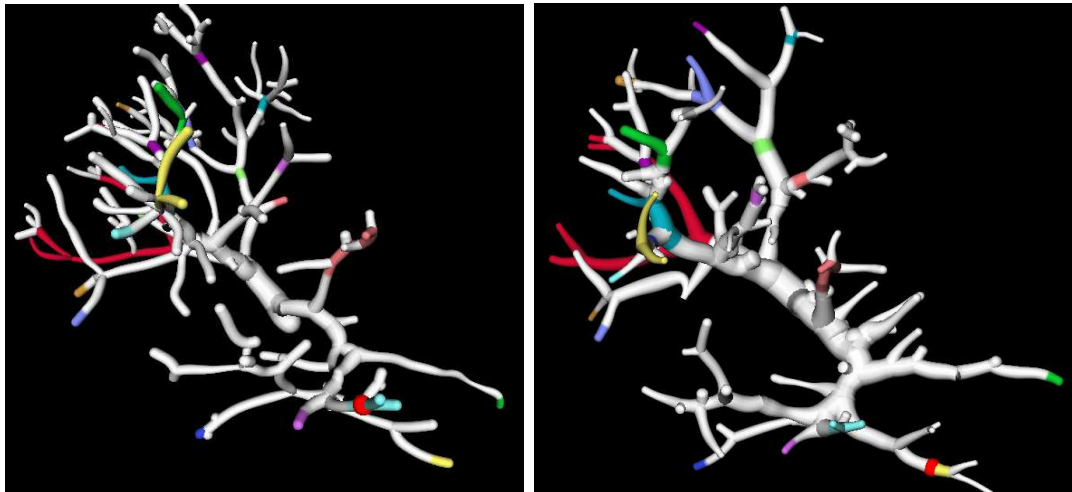


Abbildung 5.2.: Matching zweier Portalvenen

Dargestellt sind zwei Portalvenen-Bäume sowie das ermittelte Matching. Gleich eingefärbte Bereiche der Bäume sind vom Matching-Verfahren einander zugeordnet worden.

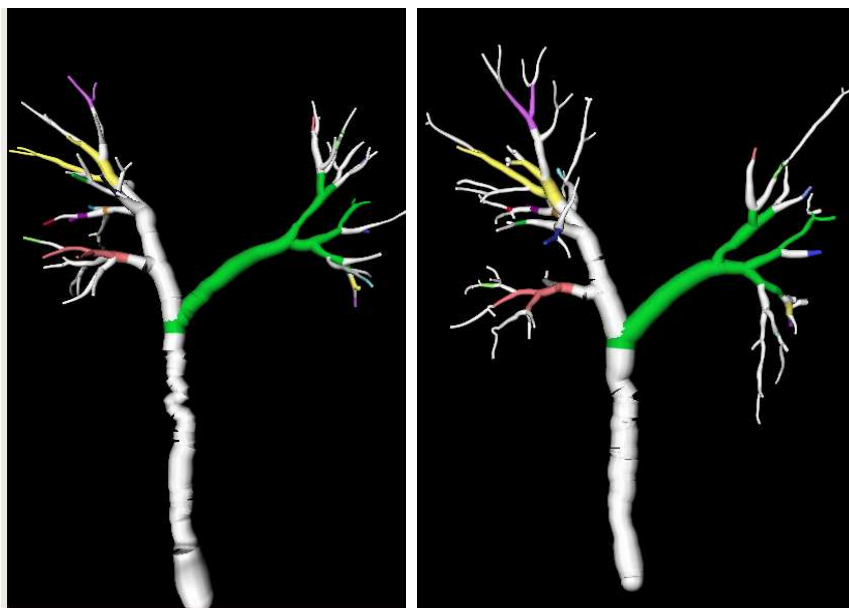


Abbildung 5.3.: Matching zweier Bronchialbäume

Dargestellt sind zwei Bronchialbäume sowie das ermittelte Matching. Gleich eingefärbte Bereiche der Bäume sind vom Matching-Verfahren einander zugeordnet worden.



## Kapitel 6.

# Registrierung

In diesem Kapitel soll ein Ausblick darauf gegeben werden, wie aus dem Matching der Gefäßbäume eine Registrierung der Bilddaten abgeleitet werden kann. Dazu definieren wir zunächst das *Registrierungsproblem*:

Sei hierfür  $d \in \mathbb{N}$  die Dimension des Bildes (in unserem Fall also  $d = 3$ ) und sei  $\Omega \subset \mathbb{R}^d$  der Definitionsbereich der Bilder. Ohne Einschränkung der Allgemeinheit sei  $\Omega = ]0, 1[^d$ . Dann definieren wir zwei Bilder  $A, B : \Omega \rightarrow \mathbb{R}$ , wobei  $A(x)$  bzw.  $B(x)$  die Intensität des jeweiligen Bildes an Position  $x$  bezeichne. Für  $x \notin \Omega$  sei  $A(x) = B(x) = 0$ .

Nach [HBHH01] berechnet eine Registrierung die räumliche Ausrichtung zweier Bilder, die nötig ist, um diese Bilder miteinander vergleichen und in dasselbe Koordinatensystem mappen zu können. Eine Registrierung setzt Objekte in einem Bild  $A$  zu Objekten in einem anderen Bild  $B$  in Beziehung und gibt eine Transformation von  $A$  nach  $B$  an. Zudem werden in der Regel die Intensitätswerte der beiden Bilder – die Voxel oder Pixel – angeglichen. Formal bezeichnet eine Registrierung also die Berechnung einer Abbildung:

$$T : x_A \longrightarrow x_B \Leftrightarrow T(x_A) = x_B$$

Der Punkt  $x_A$  in Bild  $A$  wird also durch die Transformation  $T$  auf die entsprechende Position  $x_B$  in Bild  $B$  gemappt. Die Schwierigkeit ist hierbei, die passende Transformation zu finden, die das eine Bild auf das andere ausrichtet. Man unterscheidet zwischen drei wesentlichen Transformationsarten: Den *rigiden*, den *affinen* und den *elastischen* Transformationen. Rigide Transformationen eignen sich vor allem für Aufnahmen von Objekten, die starr sind, wie zum Beispiel Knochen. Affine Transformationen sind hingegen auch für Objekte wie die Lunge, deren Volumen sich ändern kann, geeignet. Da es sich bei der Leber und Lunge jedoch um hoch-deformierbare Organe handelt, sind affine Transformationen nicht für alle Fälle ausreichend. Deswegen wird im weiteren die allgemeinste Transformationsart, eine elastische Transformation, gesucht.

Eine elastische Transformation lässt sich veranschaulichen als der Vorgang, ein aus Knetmasse bestehendes Objekt in ein anderes Objekt zu verformen und diese Verformung mit dem geringstmöglichen Aufwand zu erzielen. Jeder Verformung entspricht dabei eine Art Energieaufwand. Algorithmen, die elastische Transformationen bestimmen, suchen nach einer Transformation mit minimalem Energieaufwand, bei der ähnliche Bildbereiche aufeinander abgebildet werden. Eine ela-

stische Transformation lässt sich als *Displacement*  $u : \mathbb{R}^d \rightarrow \mathbb{R}^d$  angeben, sodass  $T(x_A) = x_B \Leftrightarrow x_A - u(x_A) = x_B$ .

Als Grundlage für die Registrierung stehen im konkreten Fall drei Informationsquellen zur Verfügung: Zum einen das ursprüngliche Bildmaterial aus den bildgebenden Verfahren, zum zweiten die segmentierten Gefäßsysteme und schließlich die Knotenzuordnungen aus dem Matching-Verfahren.

Im Folgenden werden zwei populäre Ansätze – der Darstellung in [FM03] folgend – kurz erläutert und darauf aufbauend ein Verfahren vorgeschlagen, das alle drei oben genannten Informationsquellen nutzt, um eine optimale elastische Transformation zu bestimmen.

## 6.1. Landmarkenbasierte Registrierung

Für die landmarkenbasierte Registrierung ist eine Menge von Landmarken  $a^j, b^j \in \mathbb{R}^d, j = 1, \dots, m$  gegeben. Die Bedeutung eines Paares von Landmarken  $a^j, b^j$  ist hierbei, dass der Position  $a_j$  in Bild  $A$  die Position  $b_j$  in Bild  $B$  zugeordnet werden soll; es soll also gelten:  $T(a_j) = b_j$ . In der Regel werden Landmarken durch manuelle Identifikation seitens des Benutzers bestimmt, im Kontext dieser Arbeit handelt es sich jedoch genau um die Koordinaten der zuvor gematchten Knoten. Wenn also das Matching die Zuordnung  $v_1 \leftrightarrow v_2$  geliefert hat, so wird das Landmarkenpaar  $\text{coord}(v_1), \text{coord}(v_2)$  aufgenommen.

Eine Bestimmung eines beliebigen Displacements  $u$  dergestalt, dass  $u(a^j) = a^j - b^j, j = 1, \dots, m$  gilt<sup>1</sup>, ist einfach, allerdings ist ein solches  $u$  weder eindeutig noch im Allgemeinen zufriedenstellend. Als zusätzliche Bedingung für eine gute Transformation bietet sich intuitiv an, dass diese möglichst glatt<sup>2</sup> sein sollte. Hierfür bietet sich die Einführung eines *Regularisierers*  $\mathcal{S}$  an, der die Glätte eines Displacements  $u$  bewertet. Für  $\mathcal{S}$  sind viele Alternativen bekannt, als Beispiel sei der Krümmungs-Regularisierer genannt:

$$\mathcal{S}[u] = \mathcal{S}^{\text{curv}}[u] := \frac{1}{2} \sum_{l=1}^d \int_{\Omega} (\Delta u_l)^2 dx$$

Eine optimale landmarkenbasierte Transformation bezüglich  $\mathcal{S}$  ist also ein Displacement  $u$  mit  $u(a^j) = a^j - b^j, j = 1, \dots, m$ , das  $\mathcal{S}[u]$  minimiert.

## 6.2. Intensitätsbasierte Registrierung

Bei der intensitätsbasierten Registrierung werden als Grundlage die ursprünglichen Bilddaten herangezogen. Bezüglich dieser wird ein Displacement  $u$  gesucht, sodass  $A(x - u(x)) \approx B(x)$  gilt. Es wird also ein Displacement gesucht, dass Punkte des

<sup>1</sup> $u(a^j) = a^j - b^j$  ist äquivalent zu  $T(a_j) = b_j$

<sup>2</sup>Für eine nähere Erläuterung des Begriffs *Glätte* sei auf [FM03] verwiesen. Im Rahmen dieser Arbeit soll es genügen, unter Glätte zu verstehen, dass das unter dem Displacement  $u$  entstandene Abbild des Organs keine Verzerrungen oder Brüche besitzt, also ein "natürliches" Aussehen behält.

Bildes  $A$  auf Punkte des Bildes  $B$  mit möglichst ähnlicher Intensität abbildet. Wiederum sollte dieses Displacement möglichst glatt sein, also  $\mathcal{S}[u]$  minimiert werden. Die Bedingung  $A(x - u(x)) \approx B(x)$  lässt sich mittels eines *Distanzmaßes*  $\mathcal{D}$  formalisieren, wobei  $\mathcal{D}$  von  $A, B$  und  $u$  abhängt. Eine gebräuchliche Wahl für  $\mathcal{D}$  ist

$$\mathcal{D}[A, B; u] = \frac{1}{2} \int_{\Omega} (A(x - u(x)) - B(x))^2 dx$$

Eine optimale intensitätsbasierte Transformation bezüglich  $\mathcal{S}$  und  $\mathcal{D}$  ist also ein Displacement  $u$ , das

$$\mathcal{T}[u] := \mathcal{D}[A, B; u] + \alpha \mathcal{S}[u]$$

minimiert. Hierbei ist  $\alpha$  ein Parameter, mit dem sich die Gewichtung der Glattheit des Displacements gegenüber der Bedeutung ähnlicher Bildintensitäten beeinflussen lässt.

## 6.3. Ein kombiniertes Verfahren

Die beiden in den vorhergehenden Abschnitten beschriebenen Registrierungsverfahren lassen sich leicht kombinieren. Dieses kombinierte Verfahren lässt sich interpretieren als eine Optimierung bezüglich des Intensität-Distanzmaßes und der Glattheit, die von den Landmarken geleitet wird. Das Optimierungsproblem lässt sich wie folgt formalisieren:

Minimiere  $\mathcal{T}[u] := \mathcal{D}[A, B; u] + \alpha \mathcal{S}[u]$  unter der Nebenbedingung  $u(a^j) = a^j - b^j, j = 1, \dots, m$ .

Ein Verfahren, welches ein solches optimales Displacement ermittelt, wird ebenfalls in [FM03] beschrieben. Auf diesem Wege erhält man eine optimale elastische Transformation, die die Knotenzuordnungen aus dem Matching-Verfahren und die Bildintensitäten berücksichtigt. Allerdings ist es sinnvoll, auch die segmentierten Gefäßbäume zu berücksichtigen, wie man an dem Beispiel in Abbildung 6.1 erkennt. Dieses Beispiel zeigt zwei skelettierte Gefäßsysteme. Die eingefärbten Knoten seien einander jeweils zugeordnet, d.h. ihre Positionen dienen jeweils als Landmarken. Sinnvollerweise sollten auch die rot-gefärbten Skelettlinien bei der Registrierung in Überdeckung gebracht werden. Dies wird jedoch bei der landmarkenbasierten Registrierung nicht geschehen und auch bei Berücksichtigung der Intensitäten der Bilddaten nur dann, wenn sich die Skelettlinien hinreichend deutlich abzeichnen.

Eine Berücksichtigung der segmentierten Gefäßbäume ist auf zwei Wegen denkbar:

1. Aus den Zuordnungen der "blauen" und "grünen" Knoten lässt sich im Beispiel ableiten, dass sich auch die roten Skelettlinien entsprechen sollten. Das heißt, auf diesen Skelettlinien ließen sich zusätzliche Landmarken platzieren. Beispielsweise wurden in Abbildung 6.2 auf den Kanten jeweils drei zusätzliche Landmarken gesetzt. Durch diese zusätzlichen Landmarken wird der Verlauf der Skelettlinie sehr viel besser approximiert, sodass auch das Registrierungsergebnis besser sein dürfte. Allgemein sollten so viele zusätzliche Landmarken gesetzt werden, dass der Verlauf der Skelettlinie zwischen zwei Landmarken relativ geradlinig ist. Außerdem ist natürlich zu beachten, dass jeweils gleich

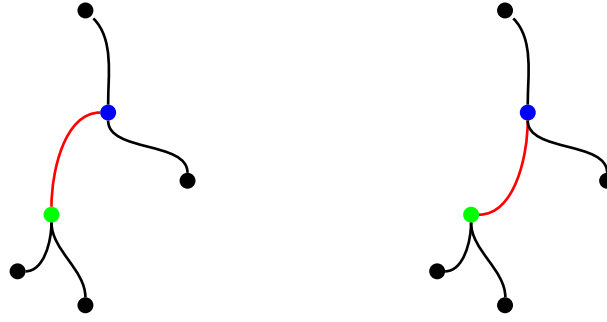


Abbildung 6.1.: Notwendigkeit der Berücksichtigung von Skelettlinien bei der Registrierung

Die Abbildung zeigt zwei skelettierte Gefäßsysteme. Die eingefärbten Knoten seien einander jeweils zugeordnet, d. h. ihre Positionen dienen jeweils als Landmarken. Sinnvollerweise sollten auch die rot-gefärbten Skelettlinien bei der Registrierung in Überdeckung gebracht werden. Dies wird jedoch bei einer einfachen landmarkenbasierten Registrierung nicht garantiert.

viele Landmarken gesetzt werden und die jeweiligen Stellen auf den beiden Skelettlinie zueinander korrespondieren müssen. Kritisch ist bei diesem Ansatz, dass die Anzahl der Landmarken deutlich erhöht wird, was die Laufzeit des Registrierungsverfahrens verschlechtern kann. Außerdem ist die Gewinnung zusätzlicher Landmarken natürlich nur für Kanten möglich, deren Endknoten zugeordnet sind. Skelettlinien, die zu unzugeordneten Kanten korrespondieren, können die Registrierung also nicht beeinflussen.

2. Alternativ kann neben der ursprünglichen Aufnahme des bildgebenden Verfahrens noch ein weiteres Bild – das Skelettlinien-Indikatorbild – derselben Größe und Auflösung erzeugt werden. Das jeweilige Voxel erhält dabei Werte im Intervall  $[0, 1]$ , wobei der Wert 1 bedeutet, dass eine Skelettlinie durch das Voxel verläuft und der Wert 0, dass keine solche durch das Voxel verläuft. Im Prinzip handelt es sich also um ein Binärbild. Da jedoch die Skelettierung des Gefäßsystems implizit mit einem Informationsverlust verbunden ist (vergleiche [Sel00]), empfiehlt es sich, eine gewisse Unschärfe beim Verlauf der Skelettlinien einzukalkulieren. Das heißt, dass Voxel, die nahe einer Skelettlinie liegen, einen Wert zwischen 0 und 1 erhalten, der umso größer ist, je näher das Voxel an einer Skelettlinie liegt. In Abbildung 6.3 ist für einen kleinen Bereich dieses Skelettlinien-Indikatorbild dargestellt. Dunklere Graustufen korrespondieren hierbei zu größeren Werten des jeweiligen Voxels. Im Registrierungsverfahren wird nun statt der Minimierung der Distanz  $\mathcal{D}[A, B; u]$  bezüglich der Intensitäten auch bezüglich der Distanz der Skelettlinien-Bilder minimiert, also:

Minimiere  $\mathcal{T}[u] := \mathcal{D}[A, B; u] + \beta \mathcal{D}[A_s, B_s; u] + \alpha \mathcal{S}[u]$  unter der Nebenbedingung  $u(a^j) = a^j - b^j, j = 1, \dots, m$ .



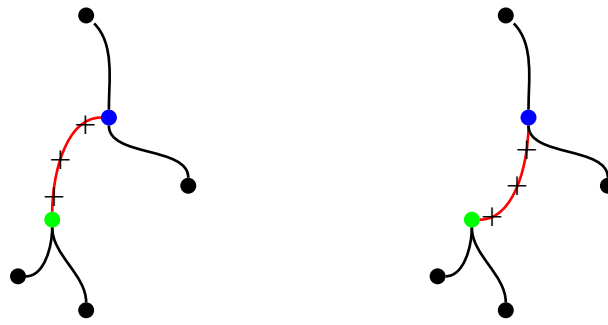


Abbildung 6.2.: Platzierung zusätzlicher Landmarken auf zugeordneten Skelettlinien

Durch die Platzierung zusätzlicher Landmarken (schwarze Kreuze) wird eine bessere Registrierung erreicht, als bei alleiniger Registrierung mittels der blauen und roten Landmarken.

Hierbei seien  $A_s$  bzw.  $B_s$  die jeweiligen Skelettlinien-Indikatorbilder und  $\beta$  ein Gewichtungsfaktor.

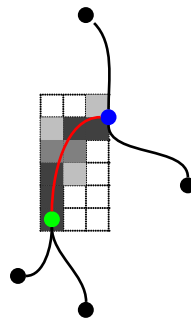


Abbildung 6.3.: Ein Indikatorbild für Skelettlinien

Mittels der Erstellung eines Indikatorbildes für Skelettlinien kann das Registrierungsergebnis gegenüber der alleinigen landmarkenbasierten Registrierung verbessert werden. Im Indikatorbild korrespondieren dunklere Graustufen zu größeren Werten des jeweiligen Voxels

Welcher der beiden Ansätze im Allgemeinen zu besseren Ergebnissen führt, bleibt zu evaluieren.

## 6.4. Beurteilung

Durch die Einbeziehung von Landmarken kann das Registrierungsergebnis gegenüber einer rein intensitätsbasierten Registrierung verbessert werden. Zu untersu-

chen bleibt, inwiefern das Registrierungsergebnis von falschen Zuordnungen des Matching-Verfahrens beeinträchtigt wird. Wie in Abschnitt 5.3 gezeigt, liefert das Matching jedoch größtenteils korrekte Ergebnisse, und die wenigen Fehler sind in der Regel nur Vertauschungen benachbarter Knoten. Daher steht zu erwarten, dass die Landmarken aus dem Matching-Verfahren hinreichend gut sind, um das Registrierungsergebnis zu verbessern. Zudem könnte es in der konkreten Situation hilfreich sein, das absolute Vertrauen des Registrierungsverfahrens in die Landmarken abzuschwächen. Wünschenswert wäre ein Verfahren, das eine Minderheit der Landmarken ignorieren darf, wenn dadurch die Glattheit der Transformation und die Übereinstimmung der Intensitäten verbessert werden kann.

# Kapitel 7.

## Ausblick

Im Rahmen dieser Diplomarbeit wurde ein Matching-Verfahren entwickelt, dass in der Lage ist, korrespondierende Stellen in typischen Gefäßsystemen aus Leber und Lunge zu identifizieren. Mit Hilfe dieses Verfahrens können im weiteren folgende Problemstellungen angegangen werden:

- Es bleibt zu untersuchen, inwiefern das Verfahren neben den typischen Gefäßsystemen auch für schwierigere Eingabedaten (wie zum Beispiel Leberregenerationsdaten) zu sinnvollen Ergebnissen führt. Hierzu wäre eine systematische Untersuchung mit Hilfe eines Ground-Truth Matchings erforderlich. Gegebenenfalls müssten die Parameter an die gegebene Situation – also erhebliche Veränderungen der zugrunde liegenden anatomischen Gefäßsysteme – angepasst werden oder auch alternative Constraints entwickelt werden.
- Wie in Kapitel 6 beschrieben, kann mit Hilfe des Matching-Ergebnisses eine Registrierung der zugrunde liegenden Bilddaten verbessert werden. Es bleibt zu untersuchen, inwieweit dieses Verfahren robust gegenüber potentiellen Fehlzugeordnungen des Matching-Verfahrens ist. Gegebenenfalls sollten bestehende landmarkenbasierte Registrierungsverfahren dahingehend modifiziert werden, dass das absolute Vertrauen des Registrierungsverfahrens in die Landmarken abgeschwächt wird (vergleiche Abschnitt 6.4). Außerdem sollte untersucht werden, welcher der beiden Ansätze zur Berücksichtigung von Skelettlinien viel versprechender ist.
- Eine weitere noch offene Frage ist, wie gut die heuristischen Verfahren zum Auffinden einer maximalen Clique sind. Unter diesen liefert das der Evolutionären Spieltheorie die besten Ergebnisse und besitzt auch für große Assoziationsgraphen eine akzeptable Laufzeit. Allerdings ist unklar, wie stark die Kardinalität der gefundenen lokal maximalen Clique von denen einer global maximalen Clique abweicht. Deswegen sollte für kleine Assoziationsgraphen eine global maximale Clique mit einem deterministischen Verfahren ermittelt werden und das Ergebnis mit denen der heuristischen Verfahren verglichen werden. Für größere Graphen ist die global maximale Clique aus prinzipiellen Gründen nicht ermittelbar.



# Anhang A.

## Graphentheorie

Im Folgenden werden einige grundlegende Begriffe aus der Graphentheorie eingeführt. Die Definitionen lehnen sich an Diestel [Die06] an. Es wird keine strenge Unterscheidung zwischen *gerichteten* und *ungerichteten* Graphen gemacht. In einem ungerichteten Graphen gilt streng genommen nicht  $E \subseteq V \times V$ , sondern  $E$  ist eine Teilmenge der zweielementigen Teilmengen von  $V$ . Der Einfachheit halber wird dies jedoch hier synonym verwendet.

**Definition 3** (Graph). *Ein Graph ist ein Paar  $G = (V, E)$  disjunkter Mengen mit  $E \subseteq V \times V$ .*

Die Elemente von  $V$  nennt man *Knoten* - oder auch *Ecken* - von  $G$ , die Elemente von  $E$  *Kanten*.  $|V|$  bezeichnet man als *Knoten kardinalität* und  $|E|$  als *Kanten kardinalität*.

Ein Knoten  $v$  und eine Kante  $e$  heißen *inzident*, wenn  $v \in e$ . Die beiden Knoten einer Kante sind ihre Endknoten, und die Kante verbindet diese Knoten. Zwei Knoten  $x, y$  eines Graphen  $G$  sind *adjazent* und heißen benachbart, wenn  $\{x, y\} \in E(G)$ . Zwei Kanten sind benachbart, wenn sie einen Endknoten gemeinsam haben.

**Definition 4** (Nachbarschaft). *Als Nachbarschaft eines Knotens  $v$ , werden alle zu  $v$  adjazenten Knoten bezeichnet:  $N(v) = \{w \in V : \exists e \in E : e = \{v, w\}\}$*

Als *Knotengrad*  $d(v)$  eines Knotens  $v$  wird die Größe seiner Nachbarschaft bezeichnet:  $d(v) = |N(v)|$ . Sind alle Paare von Knoten aus  $G = (V, E)$  benachbart, so heißt  $G$  *vollständig*.

Als *Adjazenzmatrix* eines Graphens wird eine  $|V| \times |V|$  Matrix  $A$  bezeichnet, bei der  $a_{ij} = 1$ , wenn der  $i$ . und  $j$ . Knoten miteinander verbunden sind und  $a_{ij} = 0$  sonst. Hierzu ist eine beliebige, aber feste Nummerierung der Knoten erforderlich. Bei einem ungerichteten Graphen ist  $A$  symmetrisch.

Sei im weiteren  $G' = (V', E')$  ein weiterer Graph.  $G$  heißt *isomorph* zu  $G'$ , wenn es eine Bijektion  $\varphi : V \rightarrow V'$  gibt mit  $\{x, y\} \in E \Leftrightarrow \{\varphi(x), \varphi(y)\} \in E'$  für alle  $x, y \in V$ . Eine solche Bijektion  $\varphi$  ist ein *Isomorphismus*.

Gilt  $V' \subseteq V$  und  $E' \subseteq E$ , so ist  $G' = (V', E')$  ein *Teilgraph* von  $G$  - und  $G$  ein *Obergraph* von  $G'$  - geschrieben  $G' \subseteq G$ . Ein Teilgraph  $G'$  heißt *induziert* von  $V'$  in  $G$ , wenn er alle Kanten  $\{x, y\} \in E$  enthält mit  $x, y \in V'$

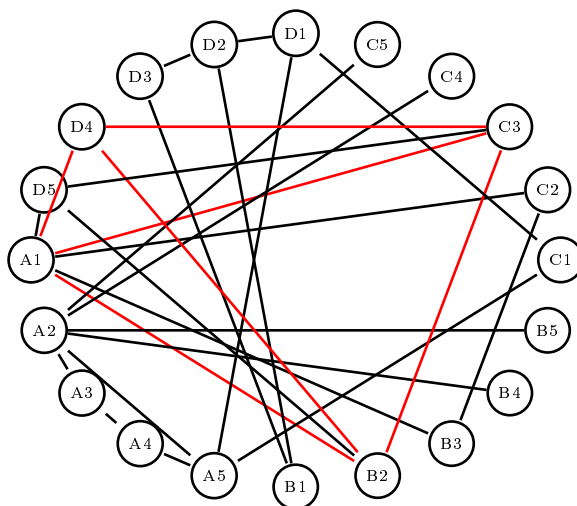


Abbildung A.1.: Ein Graph mit 25 Knoten und eine global maximale Clique (rot)

**Definition 5** (Clique). Sei  $G = (V, E)$  ein Graph. Eine Teilmenge  $Q \subseteq V$  heißt Clique, falls gilt:  $\{x, y\} \in E$  für alle  $x, y \in Q$  mit  $x \neq y$ .

Eine Clique ist also eine Teilmenge von Knoten, die paarweise in  $G$  verbunden sind. Somit ist der von  $Q$  in  $G$  induzierte Teilgraph vollständig. Eine Clique heißt *maximal*, wenn für alle  $v \in V \setminus Q$  gilt, dass  $Q \cup \{v\}$  keine Clique ist. Eine Clique  $Q$  heißt *global maximal* in  $G$ , wenn ihr Knotenkardinalität maximal unter allen Cliquen ist, wenn also für alle anderen Cliquen  $Q'$  in  $G$  gilt:  $|Q'| \leq |Q|$ .

In Abbildung A.1 ist eine global maximale Clique rot eingefärbt. Diese ist jedoch nicht eindeutig, da die Clique  $A1, B2, C3, D5$  dieselbe Kardinalität hat.

**Definition 6** (Bipartiter Graph). Sei  $G = (V, E)$  ein Graph.  $G$  heißt bipartit, wenn es nicht leere Teilmengen  $A, B \subset V$  mit  $A \cup B = V$  und  $A \cap B = \emptyset$  gibt, sodass es keine Kante  $e \in E$  gibt mit  $e \in A \times A$  oder  $e \in B \times B$ .

In einem bipartiten Graphen verlaufen also nur Kanten zwischen einem Knoten in  $A$  und einem Knoten in  $B$ , niemals zwischen zwei Knoten in  $A$  oder zwei Knoten in  $B$ .

**Definition 7** (Pfad). Ein Pfad in  $G$  ist ein nicht leerer Teilgraph  $P = (V, E)$  von  $G$  der Form

$$V = \{x_0, x_1, \dots, x_k\} \quad E = \{\{x_0, x_1\}, \{x_1, x_2\}, \dots, \{x_{k-1}, x_k\}\}$$

wobei die  $x_i$  paarweise verschieden sind. Vereinfachend schreibt man auch oft  $P = x_0 x_1 \dots x_k$ .

Die Knoten  $x_0$  und  $x_k$  sind die *Endknoten* von  $P$ , sie sind durch  $P$  verbunden. Die Anzahl der Kanten eines Pfades ist seine *Länge*. Ein nicht leerer Graph heißt *zusammenhängend*, wenn für je zwei Knoten  $x, y \in G$  mit  $x \neq y$  ein Pfad von  $x$  nach  $y$  existiert.

**Definition 8** (Kreis, Zyklus). Ist  $P = (V, E)$  mit  $P = x_0x_1\dots x_k$  ein Pfad in  $G$  mit  $k \geq 2$  und  $\{x_k, x_0\} \in G$ , so ist  $C = (V, E')$  mit  $E' = E \cup \{x_k, x_0\}$  ein Kreis in  $G$ . Vereinfachend schreibt man auch wieder  $C = x_0x_1\dots x_kx_0$ .

Ein Graph  $G$  heißt *zyklenfrei*, wenn er keine Kreise besitzt.

**Definition 9** (Baum). Ein zyklenfreier, zusammenhängender Graph  $T = (V, E)$  wird auch als Baum bezeichnet. Häufig zeichnet man einen Knoten  $w \in V$  als Wurzel des Baumes aus.

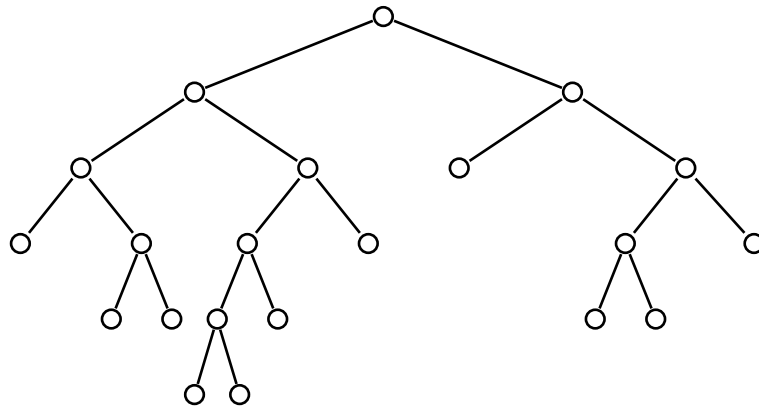


Abbildung A.2.: Ein binärer Baum

Alle Knoten  $v$  eines Baumes  $T$  mit Knotengrad  $d(v) = 1$  (mit Ausnahme der Wurzel) werden als *Blätter* bezeichnet. Zwischen zwei Knoten eines Baumes  $T$  existiert genau ein Weg in  $T$ . Als *Höhe* oder *Level* eines Knotens  $v$  bezeichnet man die Länge des (eindeutigen) Weges von der Wurzel  $w$  nach  $v$ .

Ein Baum mit Wurzel induziert automatisch eine Ausrichtung der Kanten: Als *Anfangsknoten* einer Kante bezeichnen wir denjenigen der zu ihr inzidenten Knoten, der näher an der Wurzel liegt, also ein geringeres Level hat. Der andere Knoten wird als *Endknoten* bezeichnet.

Ein Knoten, der mit einem Knoten  $v$  verbunden ist, wird als *Kind* von  $v$  bezeichnet, wenn er der Endknoten der verbindenden Kante ist.  $v$  wird dann als *Elternknoten* bezeichnet. Alle Knoten, deren eindeutiger Pfad zur Wurzel über  $v$  führt, werden als *Abkömmling* (Successor) von  $v$  bezeichnet.  $v$  wird als *Vorfahr* (Predecessor) dieser Knoten bezeichnet.

Als *topologische Distanz*  $d_t$  bezeichnet man die Anzahl der Knoten auf dem Pfad zwischen zwei Knoten, wobei man den Zielknoten mit zählt, den Startknoten jedoch nicht. Es gilt also  $d_t(v, v) = 0$  und  $d_t(v_1, v_2) = 1$ , wenn  $(v_1, v_2) \in E$ . Als *least common ancestor* (*lca*) zweier Knoten  $v_1, v_2$  bezeichnet man den eindeutig bestimmten Knoten  $v$  auf dem Weg von  $v_1$  nach  $v_2$ , der minimale Höhe besitzt. Es gilt genau dann  $lca(v_1, v_2) = v_1$ , wenn  $v_1$  Vorfahr von  $v_2$  ist.

Ein Baum, bei dem jeder Knoten maximal zwei Kindknoten besitzt wird als *binär* bezeichnet. In Abbildung A.2 ist ein binärer Baum dargestellt.



# Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Münster, 10. Juli 2006

---

Unterschrift



# Liste der Algorithmen

1.	AssocGraph()	14
2.	BinarySearch()	30
3.	GreedyClique()	44



# Abbildungsverzeichnis

1.1. Baumstruktur einer Portalvene . . . . .	3
2.1. Beispiel einer Editierfolge . . . . .	7
2.2. Ein Grundbaum des primären Deformationsmodells . . . . .	8
2.3. Ein Grundbaum des sekundären Deformationsmodells . . . . .	9
2.4. Zwei Bäume und der konstruierte bipartite Graph . . . . .	11
2.5. Mögliche Matchings von Bäumen . . . . .	12
2.6. Zwei unattributierte Bäume . . . . .	15
2.7. Assoziationsgraph, eine maximale Clique und ein Matching . . . . .	16
2.8. Assoziationsgraph und maximale Clique bei Berücksichtigung von $f_1$ . . . .	17
3.1. Pruning eines Bronchialbaums . . . . .	22
3.2. Zwei verschiedene Pfade mit ähnlicher Krümmung . . . . .	27
3.3. Zwei zu vergleichende, normierte Streckenzüge . . . . .	28
3.4. Zwei Streckenzüge mit zwei lokalen Minima für $\phi$ . . . . .	29
3.5. Beispiel für unintuitive Bewertung der Ähnlichkeit zweier Kurven . . . . .	31
3.6. Anfälligkeit der topologischen Distanz für Segmentierungsartefakte . . . . .	33
3.7. Probleme bei ungerichteter Pfadlänge . . . . .	34
3.8. Ein Hauptverzweigungsknoten . . . . .	38
5.1. Transfer einer Einfärbung der Lebersegmente nach Couinaud/Bismuth . . . .	51
5.2. Matching zweier Portalvenen . . . . .	59
5.3. Matching zweier Bronchialbäume . . . . .	59
6.1. Notwendigkeit der Berücksichtigung von Skelettlinien bei der Registrierung	64
6.2. Platzierung zusätzlicher Landmarken auf zugeordneten Skelettlinien . . . . .	65
6.3. Ein Indikatorbild für Skelettlinien . . . . .	65
A.1. Ein Graph und eine global maximale Clique . . . . .	70
A.2. Ein binärer Baum . . . . .	71



# Tabellenverzeichnis

4.1. Laufzeit und Güte der Ergebnisse der Clique-Algorithmen für einen kleinen Assoziationsgraph . . . . .	48
4.2. Laufzeit und Güte der Ergebnisse der Clique-Algorithmen für einen mittelgroßen Assoziationsgraph . . . . .	48
5.1. Vergleich unärer Constraints . . . . .	53
5.2. Kombination unärer Constraints . . . . .	54
5.3. Vergleich binärer Constraints . . . . .	55
5.4. Kombination binärer Constraints . . . . .	55
5.5. Nutzen des Prunings . . . . .	56
5.6. Wahl des Referenzknotens . . . . .	56
5.7. Vergleich der Verfahren zur Detektion einer maximalen Clique . . . . .	57
5.8. Analyse des Augmentierungsverfahrens . . . . .	57
5.9. Ergebnisse des Matchingverfahrens . . . . .	58





# Index

- 3SAT, 41
- Abkömmling, 71
- adjazent, 69
- Adjazenzmatrix, 42, 69
- Assoziationsgraph, 13, 17
- Augmentierung, 38, 57
- Baum, 19, 71
  - binärer, 72
  - Blätter von, 71
  - Pruning von, 21
  - Wurzel von, 71
- Blatt, 71
- CLIQUE, 41
- Clique, 69
  - global maximale, 14, 41, 70
  - maximale, 41, 70
- Constraint, 15, 23
  - Bewertungsfunktion für, 36
  - BifurcationAngle, 24
  - BifurcationSimilarity, 26
  - binäre, 13, 33, 54
  - CurveTangent, 35
  - EdgeDiameter, 25
  - Inheritance, 35
  - PathCurvature, 35
  - PathLength, 33
  - RN-CurveSimilarity, 27
  - RN-CurveTangent, 30
  - RN-PathCurvature, 26
  - SubtreeSimilarity, 25
  - unäre, 13, 24, 52
- Deformationsmodell
  - primäres, 8
  - sekundäres, 8
- dichotom, 22
- Distanz
  - topologische, 33, 71
- Ecke, 69
- Editierdistanz von Bäumen, 6
- Elternknoten, 71
- Genetische Algorithmen, 46
- Graph, 69
  - bipartiter, 11, 70
  - isomorphe, 69
  - Obergraph, 69
  - Teilgraph, 69
  - vollständiger, 69
  - zusammenhängender, 70
  - zyklenfreier, 71
- Graphisomorphismus, 69
- Greedy
  - Algorithmus, 43
  - Heuristik, 43
- Ground Truth, 52
- Heuristik, 42
- Individuum, 43
- inzident, 69
- Kante, 69
  - Anfangsknoten von, 71
  - Endknoten von, 71
  - inzident, 69
- Kantenkardinalität, 69
- Kindknoten, 71
- Knoten, 69
  - grad, 69
  - Abkömmling von, 71
  - adjazent, benachbart, 69
  - Höhe, Level von, 16, 71
  - inzident, 69
  - Vorfahr von, 71

- Knoten kardinalität, 69
- Kreis, 70
- least common ancestor, 15, 34, 71
- Lebersegmente
  - nach Couinaud/Bismuth, 3
- Matching, 2
  - bipartites, 11
  - Globale Ansätze, 10
  - Lokale Ansätze, 9
  - mittels eines Assoziationsgraphen, 12
  - Modellbasiertes, 7
  - Visualisierung von, 49
  - von Bäumen, 5
- Matroid, 43
- MAXIMUM CLIQUE, 41
- Motzkin-Strauss Theorem, 42
- Nachbarschaft, 69
- NP-Vollständigkeit, 41
- Obergraph, 69
- Payoff, 43
- Payoff-Matrix, 43
- Pfad, 70
  - Endknoten, 70
  - Länge, 33, 70
- Population, 43
- Pruning, 21, 55
- Referenzknoten, 24, 36, 56
- Registrierung, 61
  - affine, 61
  - Definition von, 2
  - elastische, 61
  - intensitätsbasierte, 62
  - landmarkenbasierte, 62
  - rigide, 61
- Segmentierung
  - der Gefäßsysteme, 2
- Simplex
  - Baryzentrum von, 46
  - Standard-Simplex, 42
- Spieltheorie
  - evolutionäre, 43
- Standard Replikator Gleichungen, 45
- Strategie, 43
- Support, 43
  - $\epsilon$ -Support, 46
- Teilgraph, 69
  - induzierter, 69
- Vektor
  - charakteristischer, 42, 46
- Vorfahr, 71
- Wurzel, 71
- Zyklus, 70
  - Entfernung von, 20

# Literaturverzeichnis

- [ACdB01] ARCELLI, CARLO, LUIGI P. CORDELLA und GABRIELLA SANNITI DI BAJA (Herausgeber): *Visual Form 2001, 4th International Workshop on Visual Form, IWVF-4, Capri, Italy, May 28-30, 2001, Proceedings*, Band 2059 der Reihe *Lecture Notes in Computer Science*. Springer, 2001.
- [BB76] BARROW, H. G. und R. M. BURSTALL: *Subgraph isomorphism relational structures and maximal cliques*. Information Processing Letters, 4:83–84, 1976.
- [Bil03] BILLE, PHILIP: *Tree Edit Distance, Alignment Distance and Inclusion*, 2003.
- [Bom97] BOMZE, IMMANUEL M.: *Evolution towards the Maximum Clique*. J. of Global Optimization, 10(2):143–164, 1997.
- [BV05] BRUN, LUC und MARIO VENTO (Herausgeber): *Graph-Based Representations in Pattern Recognition, 5th IAPR International Workshop, GbRPR 2005, Poitiers, France, April 11-13, 2005, Proceedings*, Band 3434 der Reihe *Lecture Notes in Computer Science*. Springer, 2005.
- [CAM<sup>+</sup>05] CHARNOZ, ARNAUD, VINCENT AGNUS, GRÉGOIRE MALANDAIN, LUC SOLER und MOHAMED TAJINE: *Tree Matching Applied to Vascular System*. In: BRUN, LUC und MARIO VENTO [BV05], Seiten 183–192.
- [Cou57] COUINAUD, C.: *Le foie – Etudes anatomiques et chirurgicales*, 1957.
- [Die06] DIESTEL, REINHARD: *Graphentheorie*. Springer-Verlag, September 2006.
- [FM03] FISCHER, BERND und JAN MODERSITZKI: *Combination of automatic non-rigid and landmark based registration: the best of both worlds*. Band 5032, Seiten 1037–1048. SPIE, 2003.
- [GH06] GRAHAM, MICHAEL W. und WILLIAM E. HIGGINS: *Globally Optimal Model-based Matching of Anatomical Trees*. Proceedings of SPIE, 6144, 2006.
- [GKK04] GERDES, I., F. KLAWONN und R. AND KRUSE: *Evolutionäre Algorithmen*, Band 1. Vieweg Verlag, 2004.
- [HBHH01] HILL, D. L. G., P. G. BATCHELOR, M. HOLDEN und D. J. HAWKES: *Medical image registration*. Physics in Medicine and Biology, 46:R1–R45, 2001.
- [HHN88] HORN, B. K. P., H. M. HILDEN und S. NEGAHDARIPOUR: *Closed-form solution of absolute orientation using orthonormal matrices*. Optical Society of America Journal A, 5:1127–1135, Juli 1988.

- [HS98] HOFBAUER, J. und K. SIGMUND: *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.
- [Kar72] KARP, R.M.: *Reducibility among combinatorial problems*. Complexity of Computer Computations, Seiten 85–103, 1972.
- [Law76] LAWLER, E.L.: *Combinatorial optimization: networks and matroids*. Holt, Rinehart & Winston, 1976.
- [Lev66] LEVENSHTAIN, VLADIMIR I.: *Binary codes capable of correcting deletions, insertions, and reversals*. Technischer Bericht 8, 1966.
- [MS65] MOTZKIN, T.S. und E.G. STRAUS: *Maxima for Graphs and a New Proof of a Theorem of Turán*. Canadian J. Math., 17:533–540, 1965.
- [Pel02] PELILLO, MARCELLO: *Matching free trees, maximal cliques, and monotone game dynamics*, 2002.
- [PSZ01] PELILLO, MARCELLO, KALEEM SIDDIQI und STEVEN W. ZUCKER: *Many-to-many Matching of Attributed Trees Using Association Graphs and Game Dynamics*. In: ARCELLI, CARLO et al. [ACdB01], Seiten 583–593.
- [Sel00] SELLE, DIRK: *Analyse von Gefäßstrukturen in medizinischen Schichtdatensätzen für die computergestützte Operationsplanung*. Shaker Verlag, Januar 2000.
- [TMP<sup>+</sup>05] TSCHIRREN, J., G. MCLENNAN, K. PALAGYI, E. A. HOFFMAN und M. SONKA: *Matching and Anatomical Labeling of Human Airway Tree*. Medical Imaging, IEEE Transactions on, 24(12):1540–1547, 2005.
- [WVA<sup>+</sup>96] WELLS, W., P. VIOLA, H. ATSUMI, S. NAKAJIMA und R. KIKINIS: *Multi-modal volume registration by maximization of mutual information*, 1996.