

Visualisierungsinstitut der Universität Stuttgart

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Studienarbeit Nr. 2407

Hierarchievisualisierung mit verallgemeinerten Pythagoras-Bäumen

Tanja Munz

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. Daniel Weiskopf
Betreuer/in:	Dr. Michael Burch

Beginn am: 2012-11-27

Beendet am: 2013-05-29

CR-Nummer: H.5.2, I.3.8

Kurzfassung

Basierend auf der Grundidee von Pythagorasbäumen wird ein neuer Ansatz zur Visualisierung hierarchischer Strukturen entwickelt. Pythagorasbäume sind Fraktale, die aus Quadraten bestehen, welche über rechtwinklige Dreiecke miteinander verbunden sind. Sie können zur Darstellung binärer Hierarchien verwendet werden, indem jedes Quadrat als Knoten aufgefasst wird. Während traditionelle Pythagorasbäume einen Verzweigungsgrad von zwei haben, werden sie hier dahingehend erweitert, dass sie auch Hierarchien mit beliebigem Verzweigungsgrad darstellen können. Dieser neue Ansatz verwendet für die Knotendarstellung Rechtecke. Als Verbindungselemente zwischen den einzelnen Knoten kommen konvexe Polygone, also nicht zwingend rechtwinklige Dreiecke wie beim traditionellen Pythagorasbaum üblich, zum Einsatz. Dabei werden verschiedene Parametereinstellungen für die visuelle Darstellung der Knoten in Bezug auf Breite, Höhe, Anordnung und Farbe untersucht. Die Visualisierungstechnik wird in einer interaktiven Anwendung bereitgestellt und ihre Nützlichkeit wird anhand von zwei Fallstudien gezeigt, in denen eine Verzeichnisstruktur und ein phylogenetischer Baum analysiert werden.

Inhaltsverzeichnis

1	Einleitung	4
1.1	Motivation	4
1.2	Ziele	5
1.3	Gliederung	5
2	Related Work	7
2.1	Node-Link-Diagramme	7
2.2	Stapelung	8
2.3	Verschachtelung	8
2.4	Einrückung	8
2.5	Fraktale	9
3	Visualisierungstechnik	11
3.1	Datenmodell	11
3.2	Traditioneller Pythagorasbaum	11
3.2.1	Grundkonstruktion	12
3.2.2	Varianten	13
3.2.3	Binäre Hierarchie	13
3.3	Verallgemeinerter Pythagorasbaum	14
3.3.1	Anordnung auf Halbkreis	15
3.3.2	Anordnung auf Kreis	17
3.3.3	Reihenfolge der Knotenanordnung	21
3.3.4	Farbkodierung	22
3.3.5	Vergleich mit Node-Link-Diagrammen	25
4	Implementierung	26
4.1	Benutzeroberfläche	26
4.2	Import	27
4.3	Funktionen und Interaktion	28
4.4	Export	31
5	Fallstudien	32
5.1	Verzeichnisstruktur	32
5.2	Phylogenetischer Baum (NCBI-Taxonomie)	37
6	Zusammenfassung und Ausblick	40
	Literaturverzeichnis	42

1 Einleitung

1.1 Motivation

Die Visualisierung von Hierarchien spielt eine große Rolle in der Informationsvisualisierung. Hierarchien kommen in sehr vielen Bereichen vor. Ein alltägliches Beispiel ist die Untersuchung von Verzeichnisstrukturen eines Computers, um eine Verteilung der Dateien oder eine Übersicht über den verbrauchten Speicherplatz zu erhalten. Aber auch in anderen Gebieten gibt es hierarchische Strukturen, wie bei der Darstellung der evolutionären Entwicklung von Organismen in phylogenetischen Bäumen, die eine Klassifizierung von Spezies liefern oder bei der Ahnenforschung in Stammbäumen. Auch Internetseiten, digitale Bibliotheken, Postleitzahlen oder durch einen Clustering-Algorithmus künstlich erzeugte Hierarchien sind entsprechend strukturiert.

Es gibt schon viele verschiedene visuelle Metaphern um Hierarchien darzustellen [JS10, Sch11]. Am bekanntesten sind Node-Link-Diagramme, die Knoten häufig als Kreise und Kanten als verbindende Linien darstellen. Als weiteres Verfahren können Tree-Maps [JS91], die eine Hierarchie durch Verschachtelung repräsentieren, genannt werden. Ferner werden Icicle Plots [KL83] durch Stapelung von Eltern- und Kinderknoten realisiert und Indented Pixel Tree Plots [BRW10] beschreiben eine Hierarchie durch Einrückung.

Im Folgenden wird eine neue Möglichkeit zur Visualisierung von Hierarchien untersucht. Diese Methode basiert auf fraktalen Strukturen, den Pythagorasbäumen [Bos57]. Das Prinzip, das diesen zu Grunde liegt, ist der Satz des Pythagoras. Ein Pythagorasbaum besteht aus Quadraten, die über rechtwinklige Dreiecke miteinander verbunden werden; der gesamte Baum kann durch Rekursion erzeugt werden. Ein solcher Baum kann zur Darstellung binärer Strukturen verwendet werden, wobei jedes Quadrat einem Knoten der Struktur entspricht. Über die rechtwinkligen Dreiecke werden hier Knoten mit ihren Kinderknoten verbunden.

Der Begriff der Fraktale wurde von Mandelbrot geprägt [Man83]. Fraktale weisen einen hohen Grad an Selbstähnlichkeit auf und werden häufig durch Rekursion gebildet. Beim Pythagorasbaum kann dadurch jedes Quadrat wieder als Stamm bzw. Wurzel eines vollständigen Baumes aufgefasst werden.

Pythagorasbäume erzeugen äußerst ansprechende Abbildungen, die an Bäume in der Natur erinnern und aufgrund ihrer Ästhetik auch in Kunstwerken verwendet werden. Jos de Mey, der für fotorealistische Darstellungen von unmöglichen Figuren bekannt ist, hat einige Bilder mit diesem Motiv gestaltet [Mey10, Ern85].

Da Pythagorasbäume immer einen Verzweigungsgrad von zwei besitzen, können mit ihnen nur binäre Hierarchien visualisiert werden. Die meisten Hierarchien, die betrachtet werden sollen, besitzen im Normalfall einen beliebigen Verzweigungsgrad.

Durch Binarisierung kann aus jedem beliebigen Baum ein binärer Baum erzeugt und dieser dann mit Hilfe eines Pythagorasbaumes dargestellt werden. Für die Transformation könnte das Verfahren von Knuth [Knu72] verwendet werden. Im Folgenden wird allerdings ein anderer Ansatz verfolgt: Eine Verallgemeinerung des Pythagorasbaumes zur Visualisierung von Hierarchien mit beliebigem Verzweigungsgrad.

Für diese Erweiterung werden die Knoten weiterhin als Quadrate dargestellt oder, verallgemeinert, in Form von Rechtecken. Die Verbindungselemente zwischen den Knoten werden zu beliebigen konvexen Polygonen – bei N Verzweigungen ein Polygon mit $N + 1$ Kanten. Im Sonderfall mit jeweils zwei Verzweigungen in der hierarchischen Struktur entsteht wieder der (traditionelle) Pythagorasbaum. Für die Knoten sind verschiedene Größen, Anordnungen und Farben in Abhängigkeit verschiedener Eigenschaften denkbar, wie z.B. der Anzahl von Blattknoten einer entsprechenden Unterhierarchie. Durch Änderung dieser Parameter kann oft eine bessere Übersicht über einen gegebenen Datensatz geliefert werden.

1.2 Ziele

Ziel dieser Arbeit ist nun die Entwicklung einer Visualisierungstechnik für beliebige Hierarchien durch Erweiterung von Pythagorasbäumen. Dabei sollen sowohl Hierarchien von Verzeichnisstrukturen verwendet, als auch aus Dateien, die im Newick-Format vorliegen, eingelesen werden können. Traditionelle Pythagorasbäume sollen erweitert werden, um diese Hierarchien mit beliebigem Verzweigungsgrad als verallgemeinerte Pythagorasbäume darzustellen. Hierbei sollen verschiedene Varianten der Abbildung für die Hierarchien getestet und verglichen werden, indem die Parameter für die Visualisierung der Knoten verändert werden. Dabei können verschiedene Faktoren Einfluss haben, zum einen auf die Form und Größe der Knoten, aber auch auf deren Position und farbliche Darstellung. Außerdem sollen Interaktionstechniken entwickelt werden um einen besseren Einblick in den Datensatz zu ermöglichen, indem in der Hierarchie navigiert werden kann und weitere Informationen über den Datensatz eingeblendet werden können. Anhand von verschiedenen Datensätzen soll in Fallstudien die Nützlichkeit der Visualisierungstechnik untersucht werden.

1.3 Gliederung

Die restliche Arbeit ist folgendermaßen gegliedert:

Kapitel 2 – Related Work: Dieses Kapitel gibt einen Überblick über bereits existierende Visualisierungstechniken für Hierarchien. Allgemeine Verfahren wie Node-Link-Diagramme, Tree-Maps, Icicle Plots und Indented Pixel Tree Plots werden vorgestellt. Zusätzlich werden Techniken, die Fraktale als Grundlage verwenden, aufgeführt.

Kapitel 3 – Visualisierungstechnik: Zunächst wird genauer beschrieben wie Pythagorasbäume konstruiert werden und dann, wie durch Erweiterung des Algorithmus Darstellungen für allgemeine Hierarchien entstehen. Verschiedene Änderungen in Bezug auf die Form der Darstellungselemente für die Knoten (Breite und Höhe), die Form der Verbindungselemente (Anordnung der Knoten auf einem Halbkreis bzw. Kreis und unterschiedliche Reihenfolge der Knoten), sowie der farblichen Kodierung werden betrachtet.

Kapitel 4 – Implementierung: In diesem Kapitel wird auf die Implementierung eingegangen. Es werden Funktionalitäten und Interaktionsmöglichkeiten der Anwendung vorgestellt.

Kapitel 5 – Fallstudien: Anhand von zwei Datensätzen wird die Anwendung und deren Visualisierungstechnik analysiert und in Bezug auf Nützlichkeit bewertet. Der erste Datensatz ist eine Verzeichnisstruktur, die aus Ordnern und Dateien besteht, der zweite ein phylogenetischer Baum, der eine Klassifizierung von Organismen enthält.

Kapitel 6 – Zusammenfassung und Ausblick: Die Arbeit wird zusammengefasst und es werden Verbesserungs- und Erweiterungsmöglichkeiten für die Anwendung und die Visualisierungstechnik geliefert.

2 Related Work

Es gibt inzwischen schon eine große Anzahl verschiedener Visualisierungstechniken um Hierarchien darzustellen. Eine Übersicht vieler existierender Techniken gibt es von Jürgensmann und Schulz [JS10]. Das Spektrum der verschiedenen Ansätze ist sehr groß und geht von Node-Link-Diagrammen über Visualisierungen mit Verschachtelungen, Stapelungen oder Einrückungen bis zu fraktalartigen Darstellungen. Dabei gibt es jeweils nochmals viele Varianten, die zweidimensional oder dreidimensional sind und sich in ihrer Art, Form und Anordnung unterscheiden. Jede der Methoden hat Vor- und Nachteile und ist nicht für alle Hierarchien gleich gut geeignet. Welche Visualisierung am besten geeignet ist, kommt auf den Einzelfall an.

Aufgrund der großen Vielfalt können im Folgenden bei weitem nicht alle Techniken aufgeführt werden. Es soll aber ein grober Überblick über die verschiedenen Ansätze geliefert werden.

Einige der Techniken werden in Abbildung 2.1 anhand einer Hierarchie, die aus sechs Knoten besteht, veranschaulicht. Zum Vergleich der aufgeführten Techniken mit der Methode, die im Nachfolgenden beschrieben wird, ist in Abbildung 2.1g ein verallgemeinerter Pythagorasbaum abgebildet.

2.1 Node-Link-Diagramme

Die verbreitetste Art, Hierarchien darzustellen, ist die Verwendung von Node-Link-Diagrammen (Abbildung 2.1a). Häufig werden dazu Knoten in Form von Kreisen dargestellt, die über Linien miteinander verbunden sind.

Hierbei gibt es viele verschiedene Varianten in Bezug auf Anordnung und Darstellung der Knoten. Häufig wird der Wurzelknoten oben positioniert und Kinderknoten darunter angeordnet.

Es gibt Untersuchungen bezüglich verschiedener Layouts der Node-Link-Diagramme [BKH⁺11]. Hier wurden sowohl traditionelle als auch orthogonale und radiale Baumlayouts betrachtet. Eine weitere Variante sind Bubble Trees [GADM06] und Balloon Drawings [LY07], bei denen jeder Unterbaum in einem Kreis eingeschlossen ist.

Der Vorteil von Node-Link-Diagrammen ist, dass die Struktur der Hierarchie klar zu erkennen ist und Pfaden einfach gefolgt werden kann. Allerdings wird der zur Verfügung stehende Platz meist nicht effizient genutzt.

2.2 Stapelung

Ein weiterer Ansatz ist die Stapelung von Knoten. Hierbei werden Kinderknoten unter bzw. auf ihre Elternknoten gesetzt.

Es gibt zwei grundlegende Ansätze: die Anordnung in einem kartesischen Koordinatensystem oder in Polarkoordinaten.

Bei Icicle Plots [KL83], wie in Abbildung 2.1b, befindet sich der Elternknoten ganz oben und die Kinderknoten werden darunter gezeichnet. Der Wurzelknoten benötigt dadurch, wie bei vielen Varianten des Node-Link-Diagramms, horizontal genau so viel Platz wie alle seine Kinderknoten zusammen.

Im zweiten Fall wird die Größe von Knoten durch einen Winkel festgelegt. Ein Beispiel ist Sunburst [SZoo] oder InterRing [YWRo2], die Ergebnisse wie in Abbildung 2.1c liefern. Die Wurzel befindet sich in der Mitte und wird als Kreis dargestellt; die Kinder werden nach außen hin aufgesetzt. Hierdurch steht den Kinderknoten mehr Platz zur Verfügung als beim kartesischen Ansatz.

Bei diesen Methoden erhält man, im Vergleich zu den Node-Link-Diagrammen, durch Veränderung der Größe der Rechtecke bzw. Winkel noch eine zusätzliche Möglichkeit zur Darstellung von Größeninformationen.

2.3 Verschachtelung

Tree-Maps [JS91] werden durch Verschachtelung erzeugt. In Abbildung 2.1d ist ein Beispiel zu sehen. Hier werden Knoten durch Rechtecke dargestellt, sodass sich Kinderknoten innerhalb ihres Elternknotens befinden. Diese Methode nutzt den Platz effizient, allerdings fällt es häufig schwer, die hierarchische Struktur genau zu erkennen.

Es gibt viele verschiedene Unterteilungsalgorithmen für die Anordnung der Knoten. Ursprünglich wurde das Slice-and-Dice-Verfahren verwendet. Bei Squarified Treemaps [BHWoo] werden Knoten angenähert durch Quadrate repräsentiert, um die Rechtecke einfacher miteinander vergleichen zu können. Anstatt von Rechtecken können auch andere Formen verwendet werden. Ein komplexeres Verfahren erzeugt Voronoi-Diagramme [BDLo5]. Kreisförmige Tree-Maps, sogenannte Pebbles [Weto3], erhält man, wenn als Grundform ein Kreis verwendet wird wie in Abbildung 2.1e. Hier wird allerdings der Platz nicht so effizient genutzt.

2.4 Einrückung

Eine weitere Methode verwendet Einrückung zur Positionierung der Kinderknoten. Alle Knoten sind auf der Horizontalen nebeneinander angeordnet, aber in verschiedenen Höhen. Knoten gleicher Tiefe werden auch in der gleichen Höhe gezeichnet. Dieses Verfahren benötigt viel horizontalen Platz. Ein Beispiel ist in Abbildung 2.1f zu sehen.

Eine vergleichbare Art der Darstellung findet man häufig in Dateimanagern, wo die Knoten vertikal angeordnet sind und interaktiv die verschiedenen Hierarchieebenen ein- und ausgeblendet werden können.

Indented Pixel Tree Plots [BRW₁₀] basieren auf dieser Methode. Sie sind auch für große und tiefe Hierarchien geeignet, um einen Überblick über die hierarchische Struktur zu erhalten.

2.5 Fraktale

Fraktale [Man83] haben die Eigenschaft der Selbstähnlichkeit und kommen häufig in der Natur vor. Ihre Verwendung zur Visualisierung von Hierarchien wurde in [KY93] untersucht. Hier wird auch eine Anwendung für dreidimensionale Hierarchievisualisierung vorgestellt. Eine weitere fraktalartige dreidimensionale Visualisierung liefern PhylloTrees [CA04]. Diese verwenden Muster, die in der Blattstellung von Pflanzen vorhanden sind, zur Erzeugung von Baumlayouts.

Sowohl DeepTree [BHP⁺₁₂] als auch OneZoom [RH12] sind durch Fraktale inspiriert. Mit diesen Visualisierungswerkzeugen kann man einen phylogenetischen Baum betrachten und interaktiv untersuchen. Allerdings muss bei OneZoom eine beliebige Hierarchie in eine binäre Struktur umgewandelt werden; DeepTree kann eine Hierarchie mit beliebigen Verzweigungen darstellen.

Es gibt weitere Ansätze, die auf Fraktalen basieren und bei denen Hierarchien nur durch binäre Bäume dargestellt werden können. Eine solche Methode, die Hierarchien als Node-Link-Diagramme darstellt, wird in [OLY05] vorgestellt.

Zufällige binäre Bäume werden in [DK96] verwendet um Bilder botanischer Bäume zu erhalten. Hierbei werden für die einzelnen Zweige eine Länge, Breite und ein Winkel in Abhängigkeit der Größe der Unterbäume für die Darstellung berechnet. Eine ähnliche Idee wird bei Botanical Trees [KWW01] verwendet, bei denen dreidimensionale Darstellungen erzeugt werden, die auch an natürliche Bäume erinnern. Die Konstruktion dieser Bäume basiert allerdings nicht auf Fraktalen.

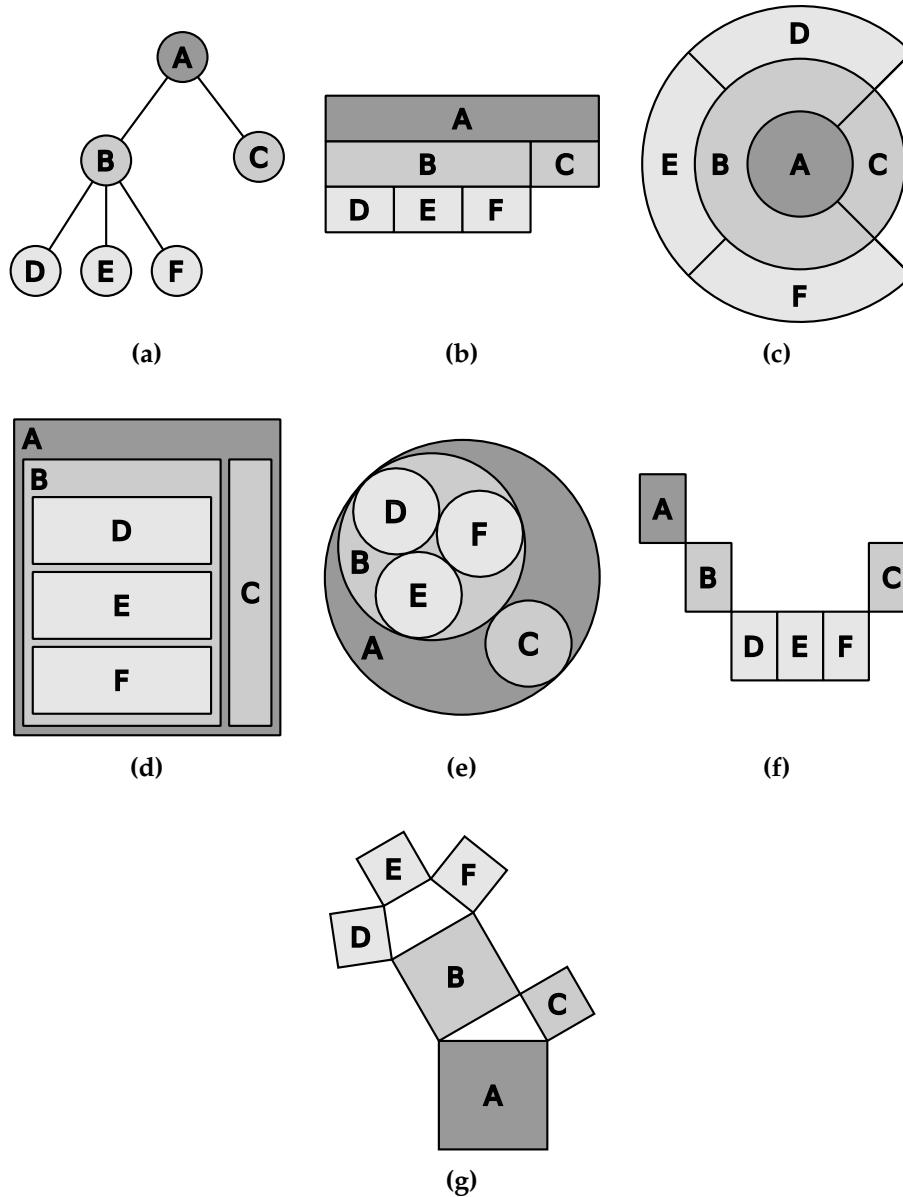


Abbildung 2.1: Veranschaulichung verschiedener visueller Metaphern für Hierarchien: (a) Node-Link-Diagramm; (b) Stapelung; (c) Sunburst; (d) Tree-Map; (e) kreisförmige Tree-Map; (f) Einrückung; (g) verallgemeinerter Pythagorasbaum.

3 Visualisierungstechnik

In diesem Kapitel wird die verwendete Visualisierungstechnik vorgestellt. Dabei wird zuerst auf die Konstruktion des traditionellen Pythagorasbaumes eingegangen, die als Grundlage für die Weiterentwicklung zum verallgemeinerten Pythagorasbaum dient. Darüber hinaus werden mögliche Varianten traditioneller Pythagorasbäume aufgeführt. Da mit diesen Bäumen nur Hierarchien mit einem Verzweigungsgrad von zwei dargestellt werden können, wird im Anschluss dazu zur verallgemeinerten Visualisierungsmöglichkeit übergegangen. Es werden unterschiedliche Varianten zur Visualisierung der gegebenen Daten vorgestellt, wobei verschiedene Eigenschaften berücksichtigt werden können, die Einfluss auf die Größe und Anordnung der Knoten haben. Anschließend wird noch das Einfärben der Knoten durch unterschiedliche Farbkodierungen beschrieben.

3.1 Datenmodell

Die verwendeten Hierarchien werden als gerichtete Graphen $H = (V, E)$ modelliert, wobei $V = \{v_1, \dots, v_k\}$ die Menge von k Knoten und $E \subset V \times V$ die Menge der Kanten repräsentiert. Die Kanten stellen die Eltern-Kind-Beziehung dar. Es gibt einen Wurzelknoten ohne eingehende Kante, die restlichen Knoten haben einen Eingangsgrad von eins. Der Ausgangsgrad jedes Knotens ist beliebig. Die Tiefe eines Knotens v' ist die Anzahl der Knoten auf dem Pfad vom Wurzelknoten v bis v' . Jedem Knoten kann ein Gewicht w zugeordnet werden, das beispielsweise für eine Größeninformation steht.

3.2 Traditioneller Pythagorasbaum

Albert E. Bosman hat 1942 Pythagorasbäume beschrieben [Bos57]. Sie wurden nach dem griechischen Mathematiker Pythagoras benannt, da sie auf dem Satz des Pythagoras basieren. Die ursprüngliche Variante dieses Baumes besteht aus Quadraten, über denen jeweils zwei weitere Quadrate an den Seiten eines rechtwinkligen Dreiecks angeordnet werden können. Es sind mehrere Varianten dieser Konstruktion denkbar. Die Winkel zwischen Hypotenuse und Katheten eines rechtwinkligen Dreiecks können beliebig variiert werden und statt rechtwinkliger Dreiecke können auch beliebige andere Dreiecke eingesetzt werden.

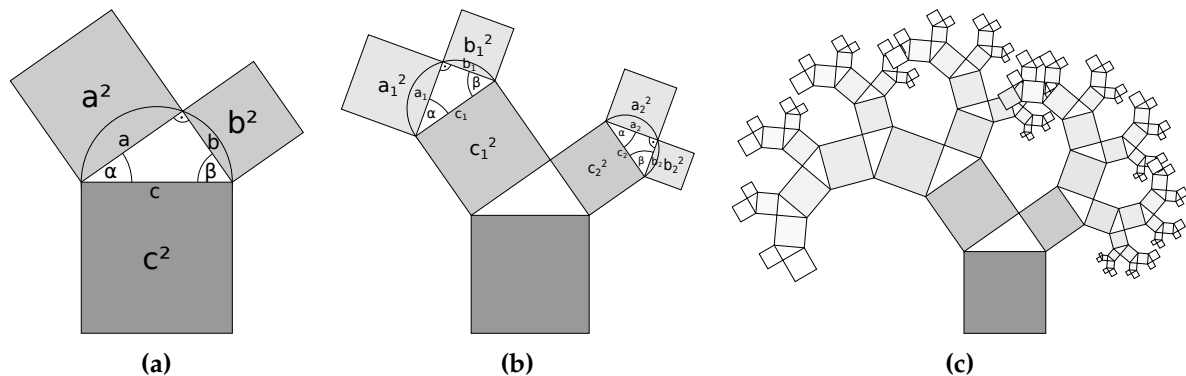


Abbildung 3.1: Veranschaulichung der rekursiven Konstruktion eines Pythagorasbaumes: (a) Grundkonstruktion für einen Iterationsschritt und Veranschaulichung des Satzes des Pythagoras; (b) mit einem weiteren Iterationsschritt; (c) Pythagorasbaum mit Tiefe sieben.

3.2.1 Grundkonstruktion

Ein Pythagorasbaum besteht aus Quadraten, die über rechtwinklige Dreiecke miteinander verbunden sind. Wie der Name schon andeutet, dient hier der Satz des Pythagoras als Grundidee für die Konstruktion: $a^2 + b^2 = c^2$.

Im Folgenden betrachten wir die Konstruktion genauer. Zunächst ist eine Grundlinie für das erste Quadrat des Pythagorasbaums gegeben. Über dieser Linie wird ein Quadrat gebildet. Die obere Kante des Quadrates dient als Hypotenuse c für ein rechtwinkliges Dreieck. Der noch fehlende Punkt für die Konstruktion des rechtwinkligen Dreiecks kann beispielsweise durch Zuhilfenahme eines Thaleskreises ermittelt werden. Die Katheten, die hierdurch gewonnen werden, dienen als Grundlinien a und b für zwei neue Quadrate. Für diese neuen Quadrate wird dieses Verfahren rekursiv bis zu einer gewünschten Tiefe wiederholt.

In Abbildung 3.1 wird die entsprechende Konstruktion für einen Pythagorasbaum veranschaulicht. Zunächst ist diese in der Teilabbildung 3.1a für einen Baum der Tiefe zwei zu sehen. Hier ist zusätzlich der Zusammenhang zum Satz des Pythagoras veranschaulicht. In Abbildung 3.1b ist eine weitere Iteration abgebildet und in Abbildung 3.1c hat der Baum bereits eine Tiefe von sieben.

Wird die Gleichung des Satzes des Pythagoras ($a^2 + b^2 = c^2$) genauer betrachtet, fällt auf, dass der Flächeninhalt des Quadrats über der Hypotenuse (c^2) dem Flächeninhalt der zwei neuen Quadrate über den Katheten ($a^2 + b^2$) entspricht. Daraus kann geschlossen werden, dass die Summe der Flächeninhalte der Quadrate bei jeder Tiefe gleich ist, da der Flächeninhalt eines Quadrats immer auf die zwei folgenden aufgeteilt wird.

Das rechtwinklige Dreieck zwischen drei Quadraten besitzt neben dem rechten Winkel die Winkel α und β , wobei $\alpha + \beta = 90^\circ$. Der Winkel α kann einen Wert zwischen 0° und 90° einnehmen. Er kann in jedem Dreieck den gleichen Wert erhalten oder zwischen den Dreiecken variieren. Die Form des Baumes kann dadurch sehr leicht verändert werden und

es können viele verschiedene Baumformen entstehen. Bereits Bosman hat mit verschiedenen Winkeln unterschiedliche überraschende Ergebnisse erhalten [Bos57]. In Abbildung 3.2 sind mehrere mögliche Darstellungen für einen Pythagorasbaum abgebildet, der jeweils eine Tiefe von elf hat. Der Baum in Abbildung 3.2a besitzt bei jedem rechtwinkligen Dreieck zwischen den Katheten und der Hypotenuse einen Winkel von 45° . Dadurch entsteht ein symmetrischer Pythagorasbaum, der an Bäume in der Natur erinnert. Werden die Winkel z.B. auf 35° und 55° gesetzt, wird eine spiralförmige Darstellung wie in Abbildung 3.2b erzeugt. Zusätzliches Vertauschen der Winkel bei jeder Tiefe wie in Abbildung 3.2c erzeugt einen Baum, der an einen Farn oder eine Fichte erinnert [PJS92]. In Abbildung 3.2d ist noch ein weiterer möglicher Baum dargestellt: die Winkel in jedem Dreieck sind zufällig gewählt. Bei dieser Methode können die unterschiedlichsten Ergebnisse entstehen, von sehr schlanken und hohen Bäumen über Bäume mit Spiralen, bis hin zu sehr verzweigten und breiten Bäumen.

3.2.2 Varianten

Weitere Varianten des Baumes entstehen, wenn ein beliebiges anderes Dreieck verwendet wird, also kein rechtwinkliges [PJS92]. Hier gilt dann nicht mehr unbedingt der Satz des Pythagoras.

Werden gleichschenklige Dreiecke verwendet, kann ein Baum wie in Abbildung 3.2f entstehen. Diese Darstellung erinnert an Brokkoli. Eine Visualisierung, die an Icicle Plots erinnert (Abbildung 3.2g) entsteht, wenn das Dreieck links und rechts einen Winkel von 0° und oben 180° besitzt. Dadurch werden die Quadrate direkt aufeinander gesetzt. In Abbildung 3.2h werden gleichseitige Dreiecke verwendet. Jedes Quadrat ist dadurch gleich groß und es entsteht eine kachelförmige Struktur.

Nicht nur die Form des Verbindungselements (hier das Dreieck) kann verändert werden, sondern auch die der Knoten selbst [Lau89, Gra94]. Anstatt der Quadrate können Linien, Rechtecke oder beliebige andere Formen verwendet werden.

3.2.3 Binäre Hierarchie

Betrachten wir im Folgenden die Quadrate nicht mehr als geometrische Objekte sondern als Knoten, kann ein Pythagorasbaum zur Visualisierung binärer Hierarchien verwendet werden. Die seitherige Konstruktion erzeugt nur vollständige binäre Bäume. Falls beliebige binäre Hierarchien dargestellt werden sollen, muss die Rekursion für die einzelnen Teilhierarchien früher abgebrochen werden. Gewichte könnten durch die Größen der Quadrate, d.h. durch Anpassung des Winkels α dargestellt werden.

In Abbildung 3.2e ist eine entsprechende zufällige binäre Hierarchie dargestellt. Die Größen der Knoten sind abhängig von der Anzahl der Blattknoten in den Unterhierarchien.

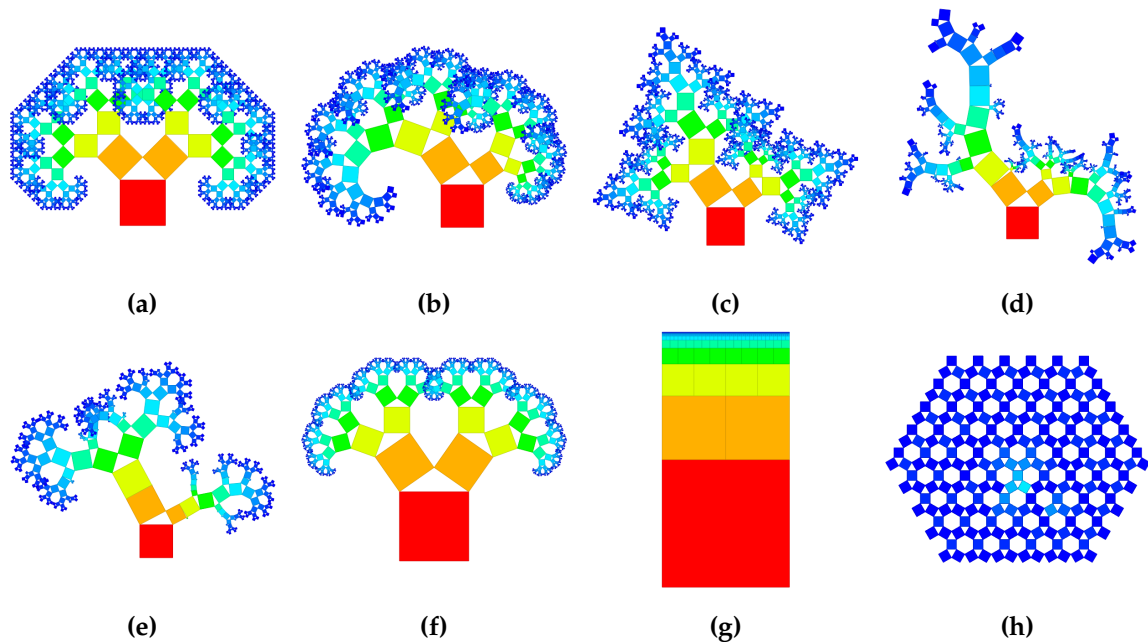


Abbildung 3.2: (Traditionelle) Pythagorasbäume mit verschiedenen Winkeln in den Dreiecken zwischen den Quadraten; bei den ersten fünf Bäumen (a - e) werden rechtwinklige Dreiecke verwendet, bei den letzten drei (f - h) gleichschenklige Dreiecke; alle Bäume stellen vollständige binäre Bäume mit einer Tiefe von elf dar, bis auf (e), welches eine zufällige binäre Hierarchie veranschaulicht. Die Winkel in den verschiedenen Darstellungen haben folgende Werte: (a) 45° auf beiden Seiten der Dreiecke; (b) 35° auf der linken und 55° auf der rechten Seite der Dreiecke; (c) 35° und 55° abwechselnd auf der rechten bzw. linken Seite; (d) verschiedene zufällige Winkel bei jedem Dreieck; (e) die Winkel sind abhängig von der Anzahl der Blattknoten in der Unterhierarchie; (f) 35° rechts und links, oben 110° ; (g) 0° rechts und links, 180° oben; (h) gleichseitiges Dreieck: bei allen Winkeln 60° .

3.3 Verallgemeinerter Pythagorasbaum

Mit dem traditionellen Pythagorasbaum können Hierarchien mit jeweils zwei Verzweigungen dargestellt werden. Durch Binarisierung kann aus jeder beliebigen Hierarchie eine binäre Struktur erzeugt und diese so visualisiert werden. Hier wird allerdings ein anderer Ansatz verfolgt, nämlich das Anpassen der Visualisierungstechnik, sodass mehrere Verzweigungen möglich sind. Dies bedeutet, dass anstatt von Dreiecken beliebige konvexe Polygone zur Verbindung eines Knotens mit seinen Kinderknoten erlaubt sind und anstatt von Quadraten verallgemeinert Rechtecke verwendet werden. Der traditionelle Pythagorasbaum ist ein Spezialfall dieses neuen Ansatzes.

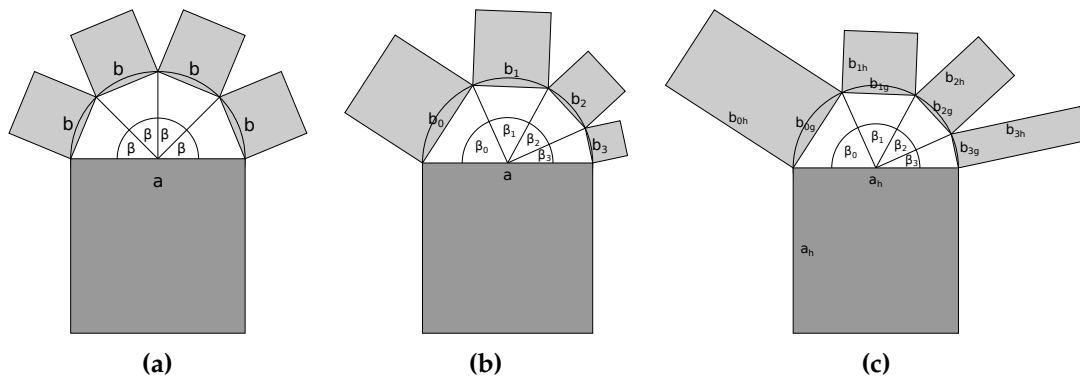


Abbildung 3.3: Konstruktion für einen verallgemeinerten Pythagorasbaum, Anordnung der Kinderknoten auf einem Halbkreis: (a) alle Kinderknoten sind gleich groß und durch Quadrate dargestellt; (b) Kinderknoten sind durch Quadrate in Abhängigkeit von Gewichten unterschiedlich groß; (c) die Kinderknoten haben unterschiedliche Breite und Höhe.

3.3.1 Anordnung auf Halbkreis

Konstruktion

Die Grundkonstruktion dieser verallgemeinerten Pythagorasbäume ist ähnlich wie bei der binären Variante. Sie entstehen auch durch Rekursion und können fraktalartig aussehen. Der Unterschied ist, dass keine Dreiecke als Verbindungselemente gebildet werden, sondern Polygone. Diese Polygone haben bei N Verzweigungen $N + 1$ Kanten: für jeden Kinderknoten und den Elternknoten jeweils eine.

Im ersten Ansatz zur Verallgemeinerung wird noch jeder Knoten, wie beim traditionellen Pythagorasbaum, als Quadrat dargestellt. Die Quadrate der Kinderknoten werden um einen Halbkreis über den Elternknoten angeordnet (vergleichbar mit dem Thaleskreis beim binären Baum). Hierfür werden die Eckpunkte der Grundlinien der neuen Quadrate auf dem Halbkreis im gleichen Abstand zueinander angeordnet. In Abbildung 3.3a ist dies grafisch veranschaulicht. Der Winkel β wird hierfür so berechnet, dass jeder Kinderknoten einen gleichen Anteil auf dem Halbkreis erhält ($\beta = \frac{180^\circ}{N}$, wobei N die Anzahl der Verzweigungen ist). Dieses Verfahren wird für jeden Kinderknoten rekursiv wiederholt.

Zur Veranschaulichung der Visualisierungstechnik wird eine Beispielhierarchie verwendet, die aus 66 Knoten besteht, mit einer maximal vorkommenden Tiefe von sieben. Jeder der Knoten besitzt eine Größeninformation. Diese Hierarchie wird in diesem und im folgenden Kapitel zur Veranschaulichung verschiedener Einstellungen und Funktionen herangezogen. Der Baum links oben in Abbildung 3.4 stellt die bis jetzt vorgestellte Methode dar.

Ein Vorteil dieser Methode ist, dass ein guter Überblick über die Verzweigungsstruktur entsteht. Allerdings werden die Knoten hier schon bei geringer Tiefe sehr klein dargestellt, dies ist besonders bei einem hohen Verzweigungsgrad der Fall. Darüber hinaus sind die

Unterbäume schlecht unterscheidbar, da sie aufgrund der fraktalen Eigenschaften sehr ähnlich aussehen. Die Tiefe der einzelnen Unterbäume ist schlecht erkennbar und es werden keine Werte bezüglich der Größen der Unterbäume dargestellt. Je tiefer der Baum ist, umso weniger ist von den tiefer gelegenen Strukturen erkennbar.

Weitere Informationen können dargestellt werden, wenn zusätzlich die Größe oder Form der Quadrate verändert wird.

Maße der Knoten

So wie es beim traditionellen Ansatz auch mehrere Varianten zur Darstellung der gleichen Hierarchie gibt, können auch hier die Verbindungselemente und Knoten unterschiedlich abgebildet werden. Sollen weitere Informationen durch die Darstellung der Knoten sichtbar werden, können anstatt der gleich großen Quadrate für Geschwisterknoten unterschiedlich große Quadrate oder Rechtecke verwendet werden. Dies bedeutet insbesondere, dass die Breite und Höhe der Knoten verändert werden. Wenn nicht jeder Kinderknoten als gleich großes Quadrat abgebildet wird, sondern in Abhängigkeit von z.B. einer Größeninformation oder der Anzahl der Blatt- oder Unterknoten eines Teilbaums, kann ein besserer Eindruck von der Hierarchie entstehen.

Breite Die Breite von Knoten kann ähnlich wie bei den traditionellen Pythagorasbäumen durch Änderung der Winkel erzielt werden. Eine entsprechende Veranschaulichung der Konstruktion ist in Abbildung 3.3b zu sehen. Durch unterschiedliche Winkel in den gleichschenkligen Dreiecken, die vom Mittelpunkt der Linie a und den Grundlinien der Quadrate der Kinderknoten gebildet werden, kann die Breite der Quadrate in Abhängigkeit von einem gewünschten Faktor festgelegt werden.

Der Winkel für einen Knoten wird in Abhängigkeit von seinem Gewicht w_i und der Summe der Gewichte seiner Geschwister, $\sum w$, ermittelt:

$$(3.1) \quad \beta_i = 180^\circ \cdot \frac{w_i}{w_i + \sum w}$$

Die Breite der Knoten ist dadurch: $b_i = a \cdot \sin\left(\frac{\beta_i}{2}\right)$

Der Einfluss der Veränderung der Quadrate ist anhand der Beispielhierarchie in Abbildung 3.4 zu sehen. Links in der Mitte wurden die Seitenlängen in Abhängigkeit von der Anzahl der Blattknoten in den Unterbäumen bestimmt und links unten in Abhängigkeit der Größeninformationen.

Bei Verwendung von unterschiedlich großen Quadraten stechen größere Teilbäume ins Auge (je nach Faktor für die Breite). Man erkennt z.B. schnell, in welcher Unterhierarchie besonders viele Blattknoten existieren. Allerdings werden kleinere Teilbäume sehr klein dargestellt und dadurch leicht übersehen bzw. sind ohne Interaktion nicht mehr erkennbar. Darüber hinaus sind einzelne Knoten über mehrere Ebenen hinweg schlecht miteinander vergleichbar, da

die Längen nicht direkt in Bezug zueinander stehen. Nur Geschwisterknoten können genau miteinander verglichen werden. Die Längen der Grundlinien sind durch die Anordnung auf einem Halbkreis nicht proportional zu den entsprechenden Winkeln. Eine Lösung wäre, den Flächeninhalt des Vaterknotens auf seine Kinderknoten zu verteilen, wie es auch beim traditionellen Pythagorasbaum geschieht ($a^2 + b^2 = c^2$). Daher wird im Folgenden zusätzlich die Höhe der Knoten verändert.

Höhe In Abbildung 3.3c ist eine allgemeine Konstruktionszeichnung für alle möglichen Anpassungen bei der Konstruktion zu sehen. Wenn die Höhe verändert werden soll, ist man nicht wie bei der Breite durch einen Halbkreis eingeschränkt. Trotzdem wird für die Ermittlung der Höhe eine Abhängigkeit von bereits existierenden Größen benötigt.

Hier gibt es mehrere Möglichkeiten wie die Höhe ermittelt werden kann. Zwei sinnvolle Grundideen sind die Aufteilung der Höhe oder des Flächeninhalts des Elternknotens auf alle Kinderknoten. Für die Berechnung der Breite und Höhe können entweder die gleichen oder unterschiedliche Gewichte verwendet werden. Im Folgenden gilt, dass w_i das Gewicht des i -ten Knotens und w das Gewicht des Elternknotens ist.

Flächeninhalt:

$$(3.2) \quad b_{ih} = a_h \cdot \frac{a_g}{b_{ig}} \cdot \frac{w_i}{w}$$

Höhe:

$$(3.3) \quad b_{ih} = a_h \cdot \frac{w_i}{w}$$

Bei beiden Varianten kann diese Aufteilung gleichmäßig auf die Kinderknoten, in Abhängigkeit der Anzahl der Blattknoten oder in Abhängigkeit der Größeninformationen eines Unterbaums erfolgen.

Die Beispielhierarchie wird auf der rechten Seite der Abbildung 3.4 nach verschiedenen Kriterien mit unterschiedlich großen Rechtecken dargestellt.

Bei einer Aufteilung des Flächeninhalts sind die Knoten in Bezug auf ihre Größe besser miteinander vergleichbar, allerdings kann die Form zweier Knoten, obwohl sie das gleiche Gewicht haben, unterschiedliche Maße haben. Bei großen bzw. tiefen Hierarchien erkennt man evtl. nichts mehr, da die Rechtecke zu schmal und lang werden und viele Überdeckungen entstehen können.

3.3.2 Anordnung auf Kreis

Wie bereits erwähnt werden beim traditionellen Pythagorasbaum die Knoten als Quadrate dargestellt und der Flächeninhalt eines Knotens entspricht der Summe der Flächeninhalte seiner Kinderknoten. Wenn man bei der verallgemeinerten Variante Rechtecke verwendet,

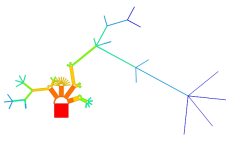
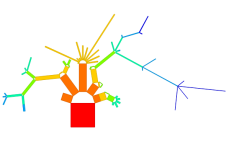
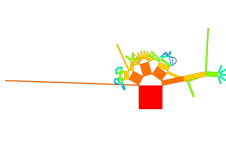
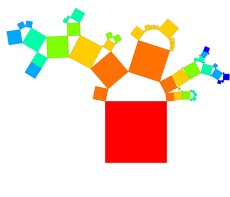
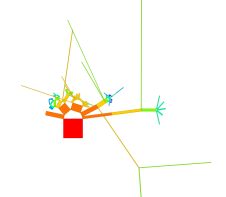
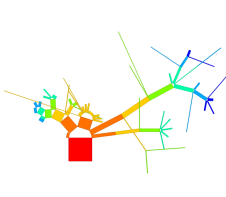
	Quadrate	Rechtecke			
		...gleichmäßig auf Kinderknoten aufgeteilt	...nach Anzahl Blattknoten aufgeteilt	...nach Größeninformation aufgeteilt	
gleiche Grundlinien		Flächeninhalt...			
		Höhe...			
Grundlinien nach Anzahl Blattknoten		Flächeninhalt...			
		Höhe...			
Grundlinien nach Größeninformationen		Flächeninhalt...			
		Höhe...			

Abbildung 3.4: Übersicht über ausgewählte Parametereinstellungen einer Beispielhierarchie. Alle Knoten werden um einen Halbkreis angeordnet. Bei den Knotenmaßen werden verschiedene Faktoren berücksichtigt.

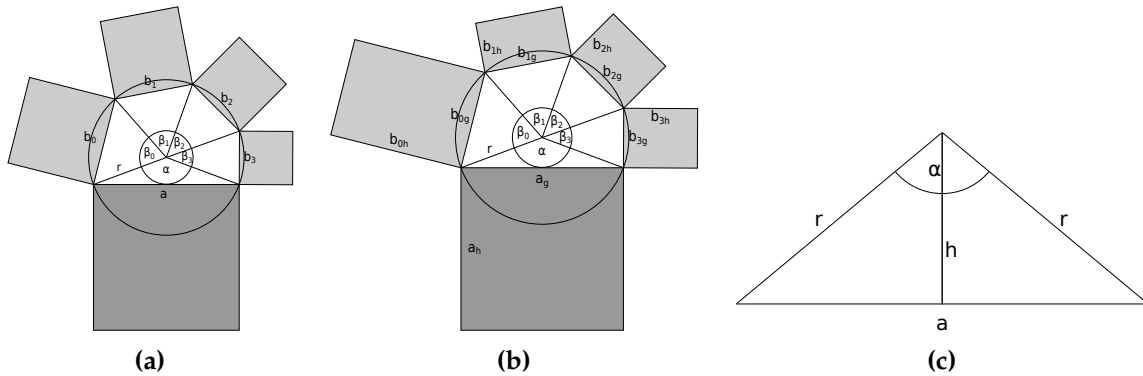


Abbildung 3.5: Anordnung der Knoten auf einem Kreis; die Summe der Flächeninhalte der Kinderknoten entspricht dem des Elternknotens: (a) Kinderknoten durch unterschiedlich große Quadrate dargestellt; (b) Kinderknoten haben unterschiedliche Maße für ihre Breite und Höhe aufgrund unterschiedlicher Gewichte; (c) Veranschaulichung des gleichschenkligen Dreiecks, das durch die obere Kante des Elternknotens und den Mittelpunkt des Kreises aufgespannt wird.

kann dies auch erzielt werden. Möchte man jedoch Quadrate mit dem gewünschten Flächeninhalt für jeden Knoten haben, wäre eine Erweiterung des Satzes des Pythagoras von $a^2 + b^2 = c^2$ (Abbildung 3.1a) zu $a^2 = b_0^2 + b_1^2 + \dots + b_{N-1}^2$ (Abbildung 3.5a) wünschenswert. Bei der neuen Formel ist a die Seitenlänge des Quadrates des Vaterknotens, b_i mit $i \in [0, N - 1]$ sind die Seitenlängen der Quadrate der Kinderknoten und N ist die Anzahl der Verzweigungen.

Konstruktion

Die Kanten für die Quadrate können nicht mehr auf einem Halbkreis angeordnet werden, außer es gibt eine oder zwei Verzweigungen. Daher muss der Bereich auf einen Kreis erweitert werden.

Das Vorgehen ist nun folgendermaßen: Zuerst werden die Längen aller Grundlinien der Kinderknoten berechnet. Diese Längen, sowie die des Vaterknotens, müssen um einen Kreis angeordnet werden, wie in Abbildung 3.5a zu sehen ist. Hierfür müssen der Mittelpunkt des Kreises und die Winkel β_i , die die Breite der Quadrate für die Kinderknoten festlegen, ermittelt werden. Danach kann wie bei der vorherigen Konstruktion fortgefahren werden.

Die einzelnen Grundlängen der Quadrate in Abhängigkeit von Gewichten (z.B. der Anzahl der Blattknoten) berechnen sich folgendermaßen:

$$(3.4) \quad b_i = a \cdot \frac{\sqrt{w_i}}{\sqrt{w}} \quad \text{mit } i \in [0, N - 1],$$

wobei w_i das Gewicht des i -ten Knotens und w das Gewicht des Elternknotens ist. Sollen alle Geschwisterknoten eine gleich lange Grundlinie erhalten, ist $b_i = \frac{a}{\sqrt{N}}$.

In den gleichschenkligen Dreiecken, die von den Grundlinien der Quadrate und dem Mittelpunkt des Kreises gebildet werden (vergleiche Abbildung 3.5c für das gleichschenklige Dreieck über dem Elternknoten), soll für den Radius Folgendes gelten:

$$(3.5) \quad r = \frac{a}{2 \cdot \sin\left(\frac{\alpha}{2}\right)} \text{ bzw. } r = \frac{b_i}{2 \cdot \sin\left(\frac{\beta_i}{2}\right)} \text{ mit } i \in [0, N-1].$$

Durch Gleichsetzen der Formeln 3.5 und Auflösen nach β_i erhält man:

$$(3.6) \quad \beta_i = 2 \cdot \arcsin\left(\sin\left(\frac{\alpha}{2}\right) \cdot \frac{b_i}{a}\right) \text{ mit } i \in [0, N-1].$$

Für die Summe der Winkel α und β_i im Kreis gilt:

$$(3.7) \quad \alpha + \sum_{i=0}^{N-1} \beta_i = 360^\circ.$$

Die Formeln 3.6 und 3.7 bilden ein Gleichungssystem mit N Gleichungen. Durch Einsetzen von 3.6 in 3.7 erhält man eine Gleichung, die mit Hilfe des Newton-Verfahrens einen Wert für α liefert:

$$(3.8) \quad \alpha + 2 \cdot \arcsin\left(\sin\left(\frac{\alpha}{2}\right) \cdot \frac{b_1}{a}\right) + \dots + 2 \cdot \arcsin\left(\sin\left(\frac{\alpha}{2}\right) \cdot \frac{b_{N-1}}{a}\right) = 360^\circ.$$

Der Kreismittelpunkt wird dann aus der Seitenlänge a , dem Winkel α und dem Radius r berechnet.

Mit Hilfe der Winkel β_i können daraufhin die Positionen der Grundlinien für die Kinderknoten auf dem Kreis bestimmt werden.

Entsprechende Darstellungen der Beispielhierarchie sind in Abbildung 3.6 zu sehen. Bei Abbildung 3.6a werden Geschwisterknoten durch gleich große Quadrate dargestellt. In Abbildung 3.6b sind die Quadrate in Abhängigkeit der Anzahl der Blattknoten abgebildet, dies bedeutet, dass alle Blattknoten gleich groß und die Elternknoten als Summe der Flächeninhalte der Blattknoten dargestellt werden. Anhand der Größe der Quadrate kann man in Abbildung 3.6c die Größeninformationen eines Unterbaums (z.B. Speicherplatzverbrauch) erkennen.

Ein Vorteil dieser Visualisierungsvariante ist, dass die einzelnen Knoten über mehrere Hierarchieebenen hinweg besser miteinander vergleichbar sind. Sowohl der Flächeninhalt, als auch die Form von Knoten, die z.B. die gleiche Anzahl an Blattknoten im Unterbaum haben, stimmen überein. Allerdings können, bei Knoten mit sehr vielen Verzweigungen, Kreise mit großen Radien entstehen, da alle Knoten nebeneinander angeordnet werden und jeweils eine geringe Höhe haben. Dies kann zu Überdeckungen und Verwirrung in Bezug auf die Zugehörigkeit der Knoten führen.

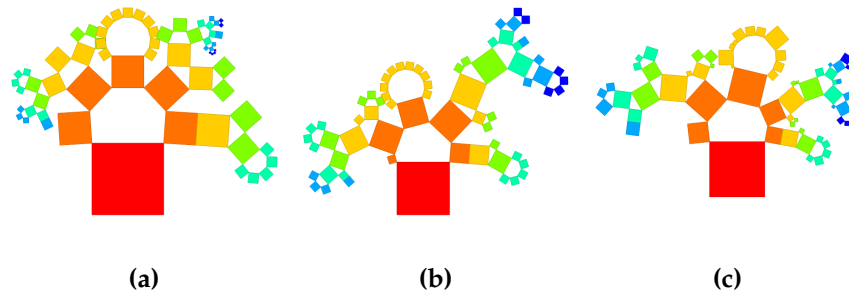


Abbildung 3.6: Anordnung der Knoten auf einem Kreis, Knotengrößen in Abhängigkeit verschiedener Faktoren: (a) gleich lange Grundlinien für Geschwisterknoten; (b) Grundlinien in Abhängigkeit von der Anzahl der Blattknoten im Unterbaum; (c) Grundlinien in Abhängigkeit der Größeninformation im Unterbaum.

Maße der Knoten

Breite Die Breite der einzelnen Quadrate ist nicht wie bei der Anordnung um einen Halbkreis beschränkt, da der Kreis in Abhängigkeit dieser Längen ermittelt wird. Die Werte dafür werden mit Formel 3.4 berechnet.

Die Winkel, die die Breite der Quadrate bestimmen, werden mit Gleichung 3.6 ermittelt.

Höhe Auch bei dieser Variante kann, wie bei der Anordnung der Knoten um einen Halbkreis, die Höhe der einzelnen Knoten verändert werden um zusätzliche Informationen darzustellen. In Abbildung 3.5b ist dies allgemein veranschaulicht. Die Berechnung dieser Werte erfolgt gleich wie im Abschnitt 3.3.1.

3.3.3 Reihenfolge der Knotenanordnung

Angelehnt an die Veränderungen, die bereits bei einem traditionellen Pythagorasbaum vorgenommen werden können, um unterschiedlichste Baumvisualisierungen zu erhalten, können auch bei der verallgemeinerten Variante noch weitere Veränderungen das Aussehen beeinflussen. Während beim binären Fall die Winkel im rechtwinkligen Dreieck verändert und vertauscht werden, wird hier die Anordnung von Geschwisterknoten auf dem (Halb-)Kreis verändert. Für die Anordnung von Geschwisterknoten existieren $N!$ Möglichkeiten. Um die Vielfalt einzuschränken, werden nur ausgewählte Anordnungen mit bestimmten Eigenschaften betrachtet. Mögliche Ansätze hierfür sind, die Knoten aufsteigend, absteigend oder je nach Tiefe abwechselnd auf- und absteigend zueinander anzuordnen. Weitere Varianten sind, dass sich der größte oder kleinste Knoten in der Mitte befindet oder große und kleine Knoten abwechselnd zueinander angeordnet sind.

Die Anordnung kann in Bezug auf verschiedene Gewichte (nach Größe, Anzahl der Blatt- oder Unterknoten) erfolgen. Je nach ausgewählter Anordnung wird die Reihenfolge, in der die Geschwisterknoten um den (Halb-)Kreis angeordnet werden sollen, variiert.

Falls die Reihenfolge aufsteigend sein soll (Abbildung 3.7c), werden die Knoten so positioniert, dass der Knoten ganz links der kleinste ist, ganz rechts der größte und dazwischen aufsteigend sortiert. Der Baum wächst so nach rechts.

Bei absteigender Reihenfolge (Abbildung 3.7e) werden die Knoten in umgekehrter Reihenfolge im Vergleich zum aufsteigenden Fall angeordnet. (Besitzt die Hierarchie bei jeder Verzweigung einen Verzweigungsgrad von ungefähr zwei, entsteht in diesem Fall eine spiralförmige Darstellung, die mit dem Pythagorasbaum in Abbildung 3.2b vergleichbar sein kann.)

Eine Kombination der beiden zuvor genannten Ansätze ist, Knoten bei einer geraden Tiefe aufsteigend und bei einer ungeraden Tiefe absteigend zu sortieren (Abbildung 3.7g).

Eine weitere Idee ist, dass der größte Knoten in der Mitte positioniert wird und nach außen die Knoten immer kleiner werden (Abbildung 3.7i). Eine sortierte Liste von Geschwisterknoten $[0, 1, 2, 3, 4, 5, 6]$ wird hier in der neuen Reihenfolge $[0, 2, 4, 6, 5, 3, 1]$ dargestellt.

Soll der kleinste Knoten in der Mitte sein, werden die Knoten so platziert, dass die größten Knoten auf den Seiten sind und die Knoten nach innen kleiner werden (Abbildung 3.7k). Diese und die vorherige Art der Anordnung erzeugen manchmal Anordnungen mit Symmetrieeigenschaften. Allerdings befinden sich dadurch eventuell sehr viele Knoten in der Mitte bzw. an den Rändern und es entstehen an den entsprechenden Stellen vermehrt Überlappungen.

Um Bereiche mit größeren Knoten nicht an fast derselben Position abzubilden, könnte das Abwechseln großer und kleiner Knoten für eine Auflockerung sorgen (Abbildung 3.7m). Die kleinsten Knoten könnten hier allerdings übersehen werden.

Je nach Anordnung kann eine Sortierung der Knoten entstehen, wodurch der kleinste oder größte Knoten schneller gefunden werden kann. Durch geschickte Anordnung können auch Überlappungen von Rechtecken bzw. Quadraten reduziert werden.

3.3.4 Farbkodierung

Seither wurde nur die geometrische Darstellung der Hierarchie betrachtet. Ein weiterer wichtiger Bestandteil für eine informative und ästhetische Visualisierung ist der sinnvolle Einsatz von Farben. Die Vergabe von Farbtönen für die Rechtecke kann auf zwei verschiedene Arten erfolgen.

Sie können in Abhängigkeit von bestimmten Werten durch Verwendung einer Farbtabelle oder durch Zuordnung zu Kategorien eingefärbt werden.

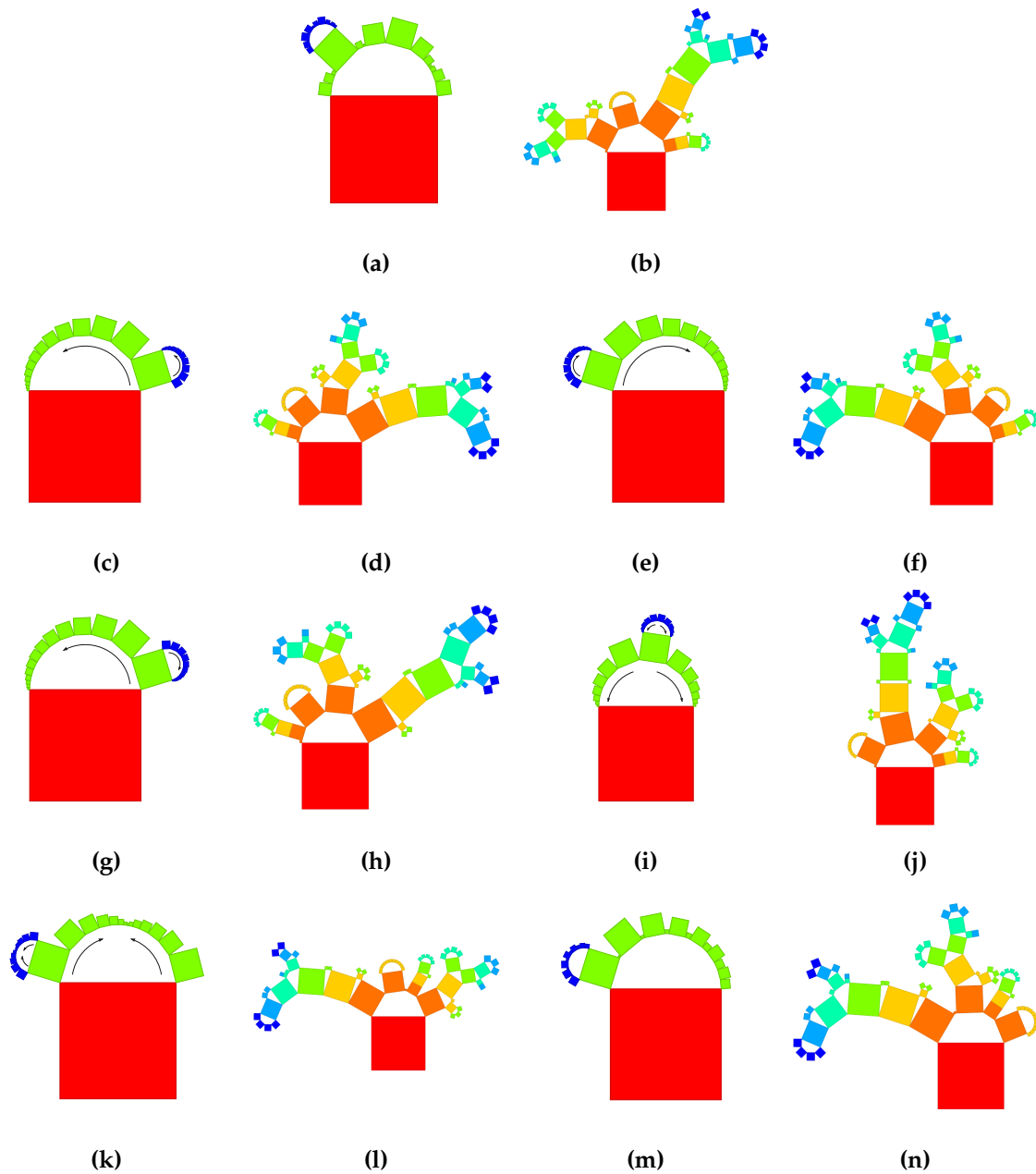


Abbildung 3.7: Verschiedene Anordnungsmöglichkeiten: schematisch und für eine Beispielhierarchie; die Pfeile deuten die Richtungen der Anordnung von großen zu kleinen Knoten an: (a, b) ungeordnet; (c, d) aufsteigend; (e, f) absteigend; (g, h) abwechselnd auf- und absteigend je nach Tiefe; (i, j) größter Knoten in der Mitte; (k, l) kleinster Knoten in der Mitte; (m, n) große und kleine Knoten abwechselnd.

Farbtabellen

Standardmäßig werden die Knoten in Abhängigkeit ihrer Tiefe in der Hierarchie eingefärbt, wie in Abbildung 3.8a. Hierdurch ist ein einfacher Vergleich der einzelnen Knoten in Bezug auf ihre Tiefe möglich und man kann schnell erkennen, wie tief die einzelnen Unterbäume sind. Das Einfärben kann aber auch nach den Größeninformationen oder in Abhängigkeit der Anzahl der Unter- bzw. Blattknoten erfolgen (Abbildung 3.8c). Diese Methode ist besonders dann nützlich, wenn die Größen der Knoten in Abhängigkeit eines anderen Faktors ermittelt wurden, um zusätzliche Informationen anzuzeigen.

Für das Einfärben wird eine Farbtabelle verwendet, die entsprechend der zu verwendenden Werte skaliert wird. Die Farbtabellen können aus beliebig vielen Farbwerten bestehen, zwischen denen interpoliert wird. Es kann sowohl linear, logarithmisch als auch exponentiell skaliert werden.

Standardmäßig wird eine Farbtabelle verwendet, die von Rot über Orange, Gelb, Grün, Türkis bis Blau geht. Es ist denkbar, jede beliebige Farbtabelle zu verwenden. In Abbildung 3.8b wird eine Farbtabelle, die von Schwarz nach Rot verläuft, zur Kodierung der Tiefeninformation verwendet.

Kategorien

Die Knoten können nach einer Häufigkeitsverteilung von Kategorien, d.h. Dateiendungen oder bestimmten Begriffen, eingefärbt werden, was Darstellungen wie in Abbildung 3.8d liefert.

Dateiendungen Für Verzeichnisstrukturen können Kategorien automatisch ermittelt werden, um eine Häufigkeitsverteilung nach Dateiendungen anzuzeigen. Hier wird bei jedem Blattknoten bestimmt, ob eine Unterteilung des Namens in einen Teil vor und nach einem Punkt möglich ist. Falls dies möglich ist, wird der Teil nach dem letzten Punkt als Dateiendung verwendet. In jedem Knoten wird dann die Häufigkeit für jede Dateiendung, die in seinem Unterbaum vorkommt, gespeichert. Im nächsten Schritt wird ermittelt, welche sechs Dateiendungen am häufigsten vorkommen und diesen unterschiedliche Farben (Rot, Grün, Blau, Cyan, Magenta und Gelb) zugewiesen. Dann wird jeder Knoten mit der Farbe der am häufigsten vorkommenden Dateiendung im Unterbaum eingefärbt. Falls diese Kategorie keine spezielle Farbe zugewiesen bekommen hat, wird der Knoten grau eingefärbt; falls gar keine Dateiendungen im Unterbaum vorkommen, weiß.

Benutzerdefinierte Kategorien Benutzerdefinierte Zeichenketten können zur Festlegung von Kategorien selbst angegeben werden, um eine Häufigkeitsverteilung zu erhalten. Anschließend wird für jeden Knoten überprüft, ob diese Zeichenketten in seinem Namen vorkommen und für jeden Knoten die Häufigkeit der Kategorien im Unterbaum gespeichert. Die Vergabe der Farben und das Einfärben der Knoten erfolgt wie bei den Dateiendungen.

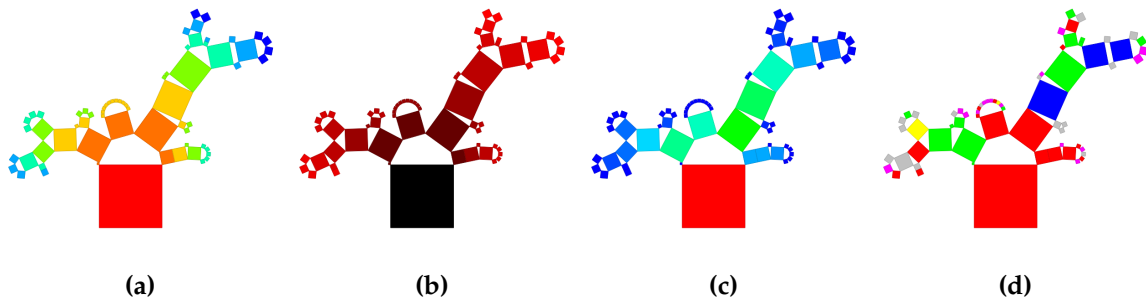


Abbildung 3.8: Unterschiedliche Farbkodierung: (a) Standard-Farbschema bezogen auf Tiefe; (b) andere Farbtabelle von Schwarz nach Rot bezogen auf Tiefe; (c) nach Anzahl der Blattknoten; (d) nach Kategorien.

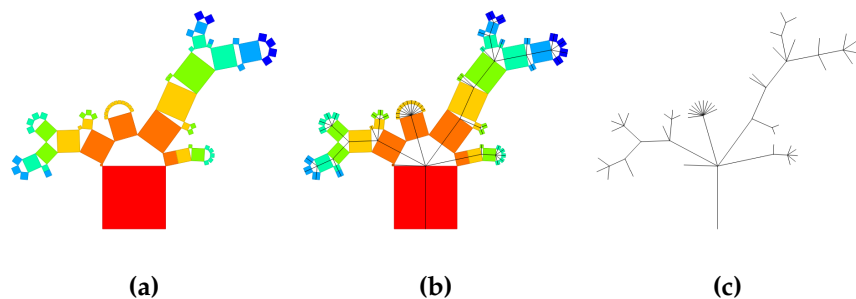


Abbildung 3.9: Node-Link-Diagramm einer Beispielhierarchie: (a) Verallgemeinerter Pythagorasbaum; (b) Verallgemeinerter Pythagorasbaum und zugehöriges Node-Link-Diagramm; (c) nur das Node-Link-Diagramm.

3.3.5 Vergleich mit Node-Link-Diagrammen

Verallgemeinerte Pythagorasbäume können auch durch äquivalente Node-Link-Diagramme dargestellt werden. In Abbildung 3.9 ist dieser Zusammenhang anhand der bereits verwendeten Beispielhierarchie abgebildet. Hierfür müssen die Positionen der Knoten nicht verändert werden. Die Mittelpunkte der Oberkanten der Rechtecke werden durch Linien miteinander verbunden. Eine Linie geht also von der Mitte der Oberkante eines Elternknotens zur Mitte der Oberkante eines Kinderknotens des verallgemeinerten Pythagorasbaumes. Diese Mittelpunkte repräsentieren im Node-Link-Diagramm die Knoten und die Linien die Kanten. Vorteile der verallgemeinerten Pythagorasbäume im Vergleich mit diesem Node-Link-Diagramm sind, dass sie den Platz effizienter nutzen und dass durch die Rechtecke (Höhe, Breite und Farbe) zusätzliche Informationen vermittelt werden können.

4 Implementierung

Die Anwendung zur Visualisierung von verallgemeinerten Pythagorasbäumen ist in C++ und Qt implementiert. Mit ihr können beliebige Hierarchien dargestellt werden, die entweder im Newick-Format vorliegen oder als Verzeichnisstruktur eingelesen werden. Dabei kann nicht nur eine Visualisierung dargestellt werden, sondern es können auch durch Änderung von Einstellungen verschiedene Parameter bezüglich der Maße, der Anordnung und des farblichen Aussehens für die Rechtecke verändert werden. Außerdem kann der Datensatz interaktiv untersucht und die resultierende Darstellung als Bild abgespeichert werden. Darüber hinaus ist es möglich, den traditionellen Pythagorasbaum (mit Verzweigungsgrad zwei) für beliebige Tiefen darzustellen und verschiedene Varianten zu untersuchen.

Alle Bilder in dieser Arbeit auf denen ein (verallgemeinerter oder traditioneller) Pythagorasbaum abgebildet ist, wurden (zumindest teilweise) mit dieser Anwendung erzeugt.

4.1 Benutzeroberfläche

Abbildung 4.1 zeigt die Benutzeroberfläche des Programms. Die Hauptbestandteile sind die Darstellungsfläche, in der die Visualisierung dargestellt wird, die Dock Widgets, die zur Veränderung von Parametern, zur Anzeige von Informationen und zur Navigation innerhalb der Hierarchie eingesetzt werden können, das Menü und die Statusleiste.

In der Darstellungsfläche wird die aktuell geladene Hierarchie dargestellt. Hier kann interaktiv mit dem Mauszeiger herausgefunden werden, welche Knoten sichtbar sind und Knoten können ausgewählt werden, um genauere Informationen über sie zu erhalten. Im sichtbaren Bereich kann gezoomt werden, der Ausschnitt kann verschoben werden und Teile der Hierarchie können hervorgehoben, ein- oder ausgeblendet werden.

In den Dock Widgets stehen Funktionen zur Veränderung der Hierarchiedarstellung zur Verfügung, sowie verschiedene Möglichkeiten zum Einfärben der Knoten. Außerdem bieten sie allgemeine Informationen zur Hierarchie sowie Navigationsmöglichkeiten, um zwischen verschiedenen Teilhierarchien zu wechseln, und eine Legende für die verwendeten Farbtöne der Knoten.

Über das Menü sind das Laden einer Hierarchie, das Exportieren der Ergebnisse sowie die Anzeige von weiteren Informationen bzw. Eigenschaften möglich. Die Statusleiste zeigt Informationen zu den Einstellungen und aktuellen Vorgängen an.

Die Anwendung kann sowohl mit einer englischen Benutzeroberfläche, als auch mit einer deutschen Übersetzung gestartet werden.

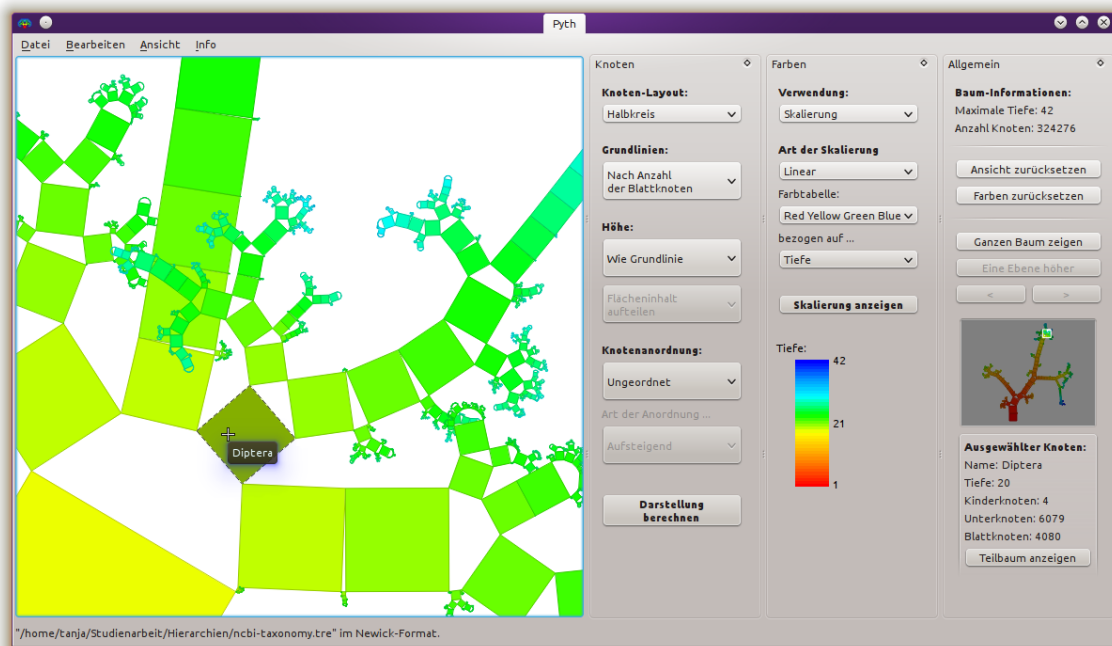


Abbildung 4.1: Benutzeroberfläche des Programms zur Darstellung von verallgemeinerten Pythagorasbäumen unter Kubuntu 12.04. Dargestellt wird ein Ausschnitt der NCBI-Taxonomie.

4.2 Import

Eine Hierarchie kann auf drei verschiedene Arten geladen werden. Um eine beliebige Hierarchie zu öffnen, gibt es zwei Möglichkeiten; binäre Baumstrukturen (für einen traditionellen Pythagorasbaum) können separat generiert werden.

Die erste Möglichkeit eine beliebige Hierarchie zu laden besteht darin, eine Hierarchie aus einer Text-Datei im Newick-Format einzulesen. Hierbei kann entweder nur die hierarchische Struktur erzeugt oder es können zusätzliche Größeninformationen für jeden Knoten berücksichtigt werden.

Eine Datei im Newick-Format kann folgendermaßen aussehen: Für jeden Knoten (bzw. dessen Bezeichner) wird seine Unterhierarchie in Klammern vorangestellt. Geschwisterknoten werden durch Kommas voneinander getrennt angegeben. Größeninformationen können nach den Bezeichnern, getrennt durch einen Doppelpunkt, angegeben werden. Ein Beispiel könnte folgendermaßen aussehen: $(B:1,(D:1)C:1)A:1$. Dieses Beispiel stellt eine Hierarchie bestehend aus vier Knoten dar. A ist der Wurzelknoten, B und C dessen Kinderknoten. C hat ein weiteres Kind, nämlich D . Hier ist jedem Knoten eine Größe von 1 zugeordnet.

Außerdem kann direkt vom Dateisystem eine Verzeichnisstruktur eingelesen werden. Hierbei kann gewählt werden, ob Größeninformationen der Dateien und Verzeichnisse (der Speicher-

verbrauch von Dateien und der Verwaltungsspeicher für einen Ordner (z.B. unter Linux)) als Größe für die Knoten verwendet werden sollen. Des Weiteren können optional versteckte Dateien berücksichtigt werden und symbolischen Links kann gefolgt werden.

Soll ein traditioneller Pythagorasbaum untersucht werden, wird eine binäre Baumstruktur erzeugt und visualisiert. Hier stehen eingeschränkte bzw. andere Funktionen zur Verfügung um den Baum zu untersuchen und um verschiedene Darstellungsvarianten zu betrachten.

4.3 Funktionen und Interaktion

Wie bereits erwähnt kann im Darstellungsbereich die Hierarchie mit verschiedenen Parametereinstellungen visualisiert werden. Hierbei ist es im allgemeinen Fall möglich, Breite, Höhe und die Reihenfolge der Knoten zu verändern und diese nach unterschiedlichen Kriterien einzufärben (wie in Kapitel 3 bereits ausführlich beschrieben).

Im binären Fall können verschiedene Varianten des traditionellen Pythagorasbaumes dargestellt werden. Hier kann eine Tiefe für die Struktur festgelegt werden sowie weitere Parameter zur Erzeugung verschiedener Baumstrukturen. Diese beinhalten eine Auswahl aus verschiedenen Arten von Dreiecken und eine Veränderung der zu verwendenden Winkel.

Zusätzliche Funktionen, Interaktions- und Navigationsmöglichkeiten sind wichtig, um die Visualisierung genauer zu untersuchen. Daher bietet die Anwendung nicht nur eine Darstellung der Hierarchie, sondern auch zusätzliche Funktionalität.

Zunächst einmal werden in der Anwendung allgemeine Informationen zur dargestellten Hierarchie angezeigt (vgl. Abbildung 4.1 rechts oben). Dies beinhaltet die Anzahl der Knoten der Hierarchie und ihre maximale Tiefe. Außerdem können zu jedem einzelnen Knoten Informationen eingeblendet werden (vgl. Abbildung 4.1 rechts unten), indem er ausgewählt wird. Diese beinhalten seinen Namen, seine Tiefe in der Hierarchie, die Anzahl seiner Kinder-, Unter- und Blattknoten und evtl. auch seine Größe und die Größe des Unterbaumes ab diesem Knoten. Wird mit dem Mauszeiger über einen Knoten gefahren, wird ein Tooltip mit dessen Namen zur schnellen Identifikation eingeblendet.

Durch Interaktion mit der Visualisierung kann man weitere Einblicke in den Datensatz erhalten. In Abbildung 4.2 werden einige Interaktionstechniken veranschaulicht, die zu einer Veränderung des Aussehens des dargestellten Baumes führen. Zum Vergleich ist die zugehörige Beispielhierarchie ohne Interaktion in Abbildung 4.2a abgebildet. Die Parameter für diese Hierarchie sind hier gezielt so gewählt, dass Überlappungen entstehen um manche Funktionalitäten besser darstellen zu können.

Die Anwendung unterstützt sowohl geometrisches als auch semantisches Zoomen [Spe07]. Beim geometrischen Zoomen wird die Darstellung der Daten nicht verändert, nur die Größe des darzustellenden Ausschnittes der gesamten Darstellung wird variiert. Beim semantischen Zoomen werden auch angezeigte Formen und Details verändert oder Objekte kommen hinzu oder verschwinden.

Hier wurde der erste Fall durch Vergrößern und Verkleinern des dargestellten Ausschnittes realisiert (Abbildung 4.2b).

Semantisches Zoomen ist so umgesetzt, dass ein Teilbaum ausgewählt werden kann und daraufhin nur dieser ohne die restlichen Knoten angezeigt wird (Abbildung 4.2c).

Falls nur ein Ausschnitt der Hierarchie angezeigt wird, kann dieser Bereich auch verschoben werden. In einer Miniaturansicht wird nochmals die gesamte Hierarchie abgebildet, wodurch eine schnelle Lokalisierung des aktuell dargestellten Bereiches ermöglicht wird. In dieser Miniaturansicht wird der nicht dargestellte Bereich gräulich eingefärbt, der sichtbare mit seinen tatsächlichen Farben.

Unterhierarchien können ein- und ausgeblendet werden (Abbildung 4.2d). Falls ein Unterbaum aggregiert wurde, wird dieser durch ein verändertes Aussehen seines Wurzelknotens dargestellt. Zum einen wird seine Farbe etwas dunkler, zum anderen erhält er eine schwarze Umrandung, deren Breite in Bezug auf die Anzahl der Unterknoten dieses Teilbaumes ermittelt wird. Da die Anzahl der Unterknoten sehr schnell abnimmt, wurde eine logarithmische Skalierung gewählt, sodass die Umrandung auch noch bei Knoten mit wenigen Unterknoten sichtbar ist.

Teile der Hierarchie können farblich hervorgehoben werden. In Abbildung 4.2e ist der Pfad von einem Knoten bis zur Wurzel schwarz mit grüner Umrandung eingefärbt. Abbildung 4.2f zeigt einen Unterbaum in Schwarz mit gelber Umrandung, um ihn hervorzuheben. Des Weiteren kann ein Unterbaum in den Vordergrund gesetzt werden (Abbildung 4.2g). Dies ist besonders nützlich, falls starke Überdeckungen durch andere Knoten auftreten und dadurch manche Knoten nicht sichtbar sind.

Wird nur ein Teilbaum angezeigt, stehen zusätzliche Navigationsmöglichkeiten zur Verfügung. In der Hierarchie kann eine Ebene aufgestiegen werden, d.h. der Teilbaum des Vaterknotens des aktuell dargestellten Wurzelknotens wird angezeigt, und es kann zurück zu einem bereits angezeigten Teilbaum navigiert werden.

Als zusätzliches Feature können für die Knoten Labels eingeblendet werden (Abbildung 4.2h). Ob Labels angezeigt werden können ist abhängig von der Knotengröße und von Überlappungen mit anderen Knoten. Zudem ist die Größe des Labels ausschlaggebend, ob der gesamte Name des Knotens oder nur ein Teil davon angezeigt wird. Da beim Zoomen und Verschieben der Ansicht evtl. mehr Platz für die Anzeige von Labels einzelner Knoten zur Verfügung steht, können hier neue, zuvor noch nicht sichtbare Labels angezeigt bzw. vollständig eingeblendet werden (Abbildung 4.3).

Außerdem gibt es eine Suchfunktion. Hier kann nach einem festgelegten Suchbegriff in der gesamten Hierarchie gesucht werden. Es kann optional festgelegt werden, ob Groß- und Kleinschreibung beachtet und ob lediglich nach Einträgen, die exakt mit dem Suchbegriff übereinstimmen, gesucht werden soll. Als Ergebnis werden alle Knoten, die den Suchbegriff enthalten, rot eingefärbt (Abbildung 4.2i) und die Anzahl der Funde wird ermittelt. Hier ist es teilweise sehr schwer, eingefärbte Knoten zu finden, die sehr klein sind. Weitere Interaktionstechniken wären nötig, um zu den Ergebnissen in der Visualisierung zu gelangen, da sonst viele Fundstellen untergehen könnten.

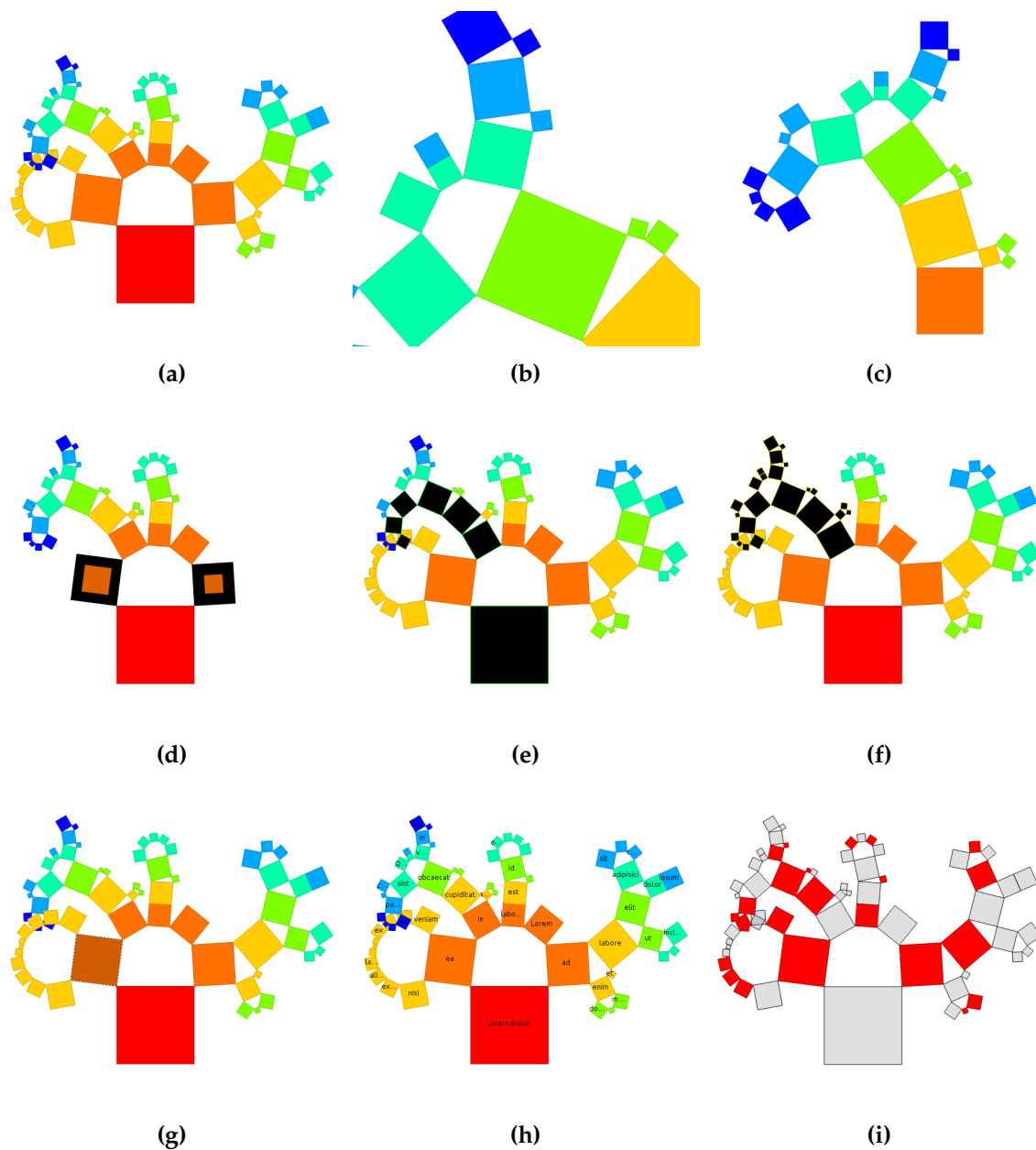


Abbildung 4.2: Veranschaulichung verschiedener Interaktionsmöglichkeiten anhand einer Beispielhierarchie: (a) ohne Interaktion; (b) Zoomfunktion; (c) Anzeige eines Teilbaums; (d) Teilbäume sind aggregiert; (e) Pfad eines Knotens ist farblich markiert; (f) Unterbaum eines Knotens ist farblich markiert; (g) Unterbaum eines Knotens ist im Vordergrund; (h) Anzeige von Labels; (i) Suche nach einer bestimmten Zeichenkette.

5 Fallstudien

Anhand zweier Datensätze wird die Nützlichkeit der Visualisierungstechnik gezeigt. Es werden hierfür verschiedene Parameter für die Visualisierung der Datensätze variiert, sowie Funktionen und Interaktionstechniken des Programms verwendet, um die Hierarchien weiter zu untersuchen. Der erste Datensatz ist eine Verzeichnisstruktur mit Größeninformationen. Hierfür wird der Quellcode des Qt Creators verwendet, mit dessen Hilfe die Anwendung programmiert wurde. Der zweite Datensatz ist ein phylogenetischer Baum, der eine Übersicht über alle lebenden Organismen auf der Erde liefert. Er liegt im Newick-Format vor.

5.1 Verzeichnisstruktur

Als erstes wird die hierarchische Anordnung einer Verzeichnisstruktur genauer betrachtet. Anhand des Quellcodes für den Qt Creator in Version 2.4.1 [qtc12] wird demonstriert, wie das Programm verwendet werden kann, um Informationen über die Verzeichnisse und Dateien zu erhalten. Der Qt Creator [qtp] ist eine Entwicklungsumgebung für die Erstellung von Anwendungen in C++ und Qt. Mit seiner Hilfe wurde die beschriebene Anwendung programmiert. Der Creator ist auch in C++ und Qt implementiert; sein Quelltext enthält daher viele cpp- und h-Dateien. Zudem beinhaltet die Verzeichnisstruktur Übersetzungen, Tests, Dokumentation und Weiteres.

Die Verzeichnisstruktur wird vom Programm direkt eingelesen. Dabei werden Größeninformationen (Speicherplatz) für die Knoten ermittelt und versteckte Dateien berücksichtigt. Das Verzeichnis hat eine Gesamtgröße von 83,1 MiB und besteht insgesamt aus 9302 Knoten. Davon sind 862 innere Knoten und 8440 Blattknoten. Jeder innere Knoten entspricht einem Verzeichnis und jeder Blattknoten einer Datei (bzw. einem leeren Verzeichnis). Die Struktur hat eine maximale Tiefe von 10 und der maximal vorkommende Verzweigungsgrad eines Knotens beträgt 378 (es handelt sich hierbei um einen Ordner der Dokumentation, der Bilder enthält (*doc/images/*)).

Interessante Fragestellungen in Bezug auf Verzeichnisstrukturen könnten sich darauf beziehen, wie die Speicherbelegung verteilt ist und wo die meisten Dateien abgespeichert sind. Aber auch welche Arten von Dateien abgespeichert sind, wo sich bestimmte Dateien befinden und wie innerhalb der Verzeichnisstruktur sinnvoll navigiert werden kann, sind notwendige Fragestellungen um den Datensatz genauer zu analysieren. Im Nachfolgenden werden für diese Problematiken Lösungsstrategien vorgestellt.

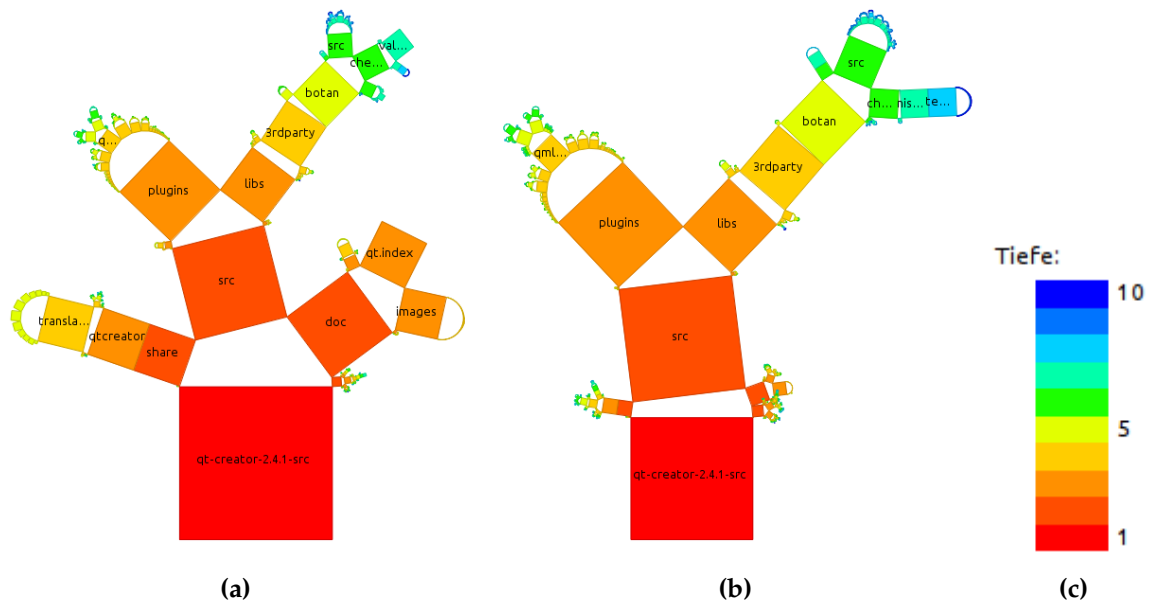


Abbildung 5.1: Visualisierungen für den Quellcode des Qt Creators in Version 2.4.1: (a) Knotengrößen in Abhängigkeit der Größeninformationen (Speicherplatz), (b) Knotengrößen in Abhängigkeit der Anzahl der Blattknoten (Dateien) und (c) das verwendete Farbschema.

In diesem Abschnitt wird in den Visualisierungen für jeden ausreichend großen Knoten ein Label angezeigt, um eine bessere Zuordnung der Knoten zu den Verzeichnissen und Dateien zu ermöglichen.

Zunächst werden für diesen Datensatz verschiedene Visualisierungen getestet. In Abbildung 5.1 sind mögliche Darstellungen abgebildet, in denen die Größe der Knoten in Abhängigkeit der Anzahl der Blattknoten und in Abhängigkeit des Speicherplatzverbrauches dargestellt wird. Beim Speicherplatzverbrauch wird für jeden Knoten ein Gewicht verwendet, das aus der Summe der Datei- und Verzeichnisgrößen im entsprechenden Unterbaum besteht. Die Knoten sind in Abhängigkeit ihrer Tiefe eingefärbt, unter Verwendung des Farbschemas in Abbildung 5.1c.

Falls die Darstellung nach Größeninformationen erfolgt (Abbildung 5.1a) ist der Speicherplatzverbrauch schnell erkennbar. Der Ordner *src* benötigt am meisten Speicher, gefolgt von *doc* und *share*. Die restlichen Verzeichnisse und Dateien dieser Hierarchieebene benötigen recht wenig Speicherplatz im Verhältnis zu diesen.

Bei der Darstellung nach der Anzahl der Blattknoten in den Unterhierarchien (Abbildung 5.1b) ist erkennbar, wo die meisten Dateien abgespeichert sind. Die Verteilung ist hier genauso wie beim Speicherplatz. Die Ordner *doc* und *share* haben aber dennoch sehr wenige Blattknoten, während der Speicherplatzverbrauch recht groß ist.

Abbildung 5.2 zeigt Darstellungen, bei denen der Flächeninhalt eines Knotens jeweils der Summe der Flächeninhalte seiner Kinderknoten entspricht. Durch diese Visualisierungen

entstehen zwar mehr Überlappungen, die für Unklarheit sorgen können, Knoten sind allerdings über verschiedene Ebenen hinweg miteinander vergleichbar. Bei Abbildung 5.1a war nur ein Vergleich von Geschwisterknoten sinnvoll. In Abbildung 5.2a und 5.2b ist nun der Speicherplatzverbrauch der einzelnen Dateien und Verzeichnisse im Verhältnis zueinander erkennbar. In Abbildung 5.2a ist die Breite der einzelnen Knoten gleich wie in Abbildung 5.1a. Durch Änderung der Höhe wird der gewünschte Flächeninhalt erzielt. In Abbildung 5.2b wird jeder Knoten durch ein Quadrat dargestellt, wobei der Flächeninhalt der Knoten gleich wie in Abbildung 5.2a ist. Hier haben Knoten mit gleichen Größeninformationen nicht nur den gleichen Flächeninhalt, sondern auch die gleiche Form. Ferner ist die Visualisierung deutlich überschaubarer, da weniger Überlappungen entstehen.

Es ist erkennbar, dass die Rechtecke für *share* und *libs* etwa gleich groß sind. Bei Abbildung 5.1a könnte irrtümlicherweise vermutet werden, dass das Verzeichnis *libs* größer als *share* ist, obwohl es in Wirklichkeit sogar kleiner ist. In Abbildung 5.2c werden die Knoten noch zusätzlich nach ihrem Speicherplatzverbrauch eingefärbt. Die genaue Speicherbelegung kann mit Hilfe der Legende (Abbildung 5.2d) den Visualisierungen entnommen werden.

Wird nach bestimmten Dateien oder Ordnern gesucht, ist eine Suchfunktion sehr nützlich. Diese wird mit dem Suchbegriff „qt“ getestet. Das Ergebnis der Suche ist in Abbildung 5.3a dargestellt. Der Suchbegriff wurde 676 Mal gefunden. Alle Knoten, die ihn enthalten, sind rot eingefärbt; allerdings sind nicht alle eingefärbten Knoten zu sehen, da sie teilweise zu klein dargestellt werden und dadurch untergehen. Dessen ungeachtet sind mehrere Ordner sichtbar, die sehr viele Dateien mit dieser Zeichenkette enthalten. Als Beispiel ist der *translation*-Ordner (auf der linken Seite) zu nennen. Hier befinden sich die Übersetzungsdateien für verschiedene Sprachen, deren Dateinamen alle nach dem gleichen Prinzip aufgebaut sind und alle die Zeichenkette „qtcreator“ enthalten.

Des Weiteren wird in Abbildung 5.3b und 5.3c eine Häufigkeitsverteilung nach Dateien und Ordnern gezeigt. Die Legende ist nach Häufigkeit geordnet. Wie ihr entnommen werden kann, kommen hier h-Dateien am häufigsten vor, gefolgt von cpp-Dateien. Die Kategorie des Wurzelknotens repräsentiert die Dateiart, die am häufigsten vorkommt. Bei allen Knoten spiegelt die Farbe immer die Dateiart wieder, die in dem entsprechenden Teilbaum am häufigsten vorkommt. Hier entsteht ein guter Überblick, wo die verschiedenen Dateiarten abgelegt wurden. Crt-Dateien (hier gelb dargestellt) befinden sich fast alle in einem Verzeichnis, während sich cpp-, h- und png-Dateien in vielen verschiedenen Ordnern befinden.

Kommen wir noch zur Navigation innerhalb der Hierarchie: Wenn Labels für die Knoten angezeigt werden, entsteht durch Zoomen oder durch Anzeige eines Teilbaums eine detailliertere Ansicht der Datei-/Verzeichnisnamen, die tiefer in der Hierarchie vorkommen. Dies ermöglicht eine Navigation ähnlich der in einem Dateimanager. Auch dort wird ein Verzeichnis ausgewählt und eine Auflistung der darin enthaltenen Dateien angezeigt. Zusätzlich liefert diese Visualisierung noch Informationen über die tiefer liegenden Inhalte in der Hierarchie, sowie evtl. Informationen über die Anzahl der Blattknoten und den Speicherplatzverbrauch.

In Abbildung 5.4 ist ein exemplarisches Beispiel für die Navigation in der Verzeichnisstruktur anhand von zwei unterschiedlichen Navigationszielen veranschaulicht. Hier wird durch sowohl geometrisches als auch semantisches Zoomen zu unterschiedlichen Dateien navigiert.

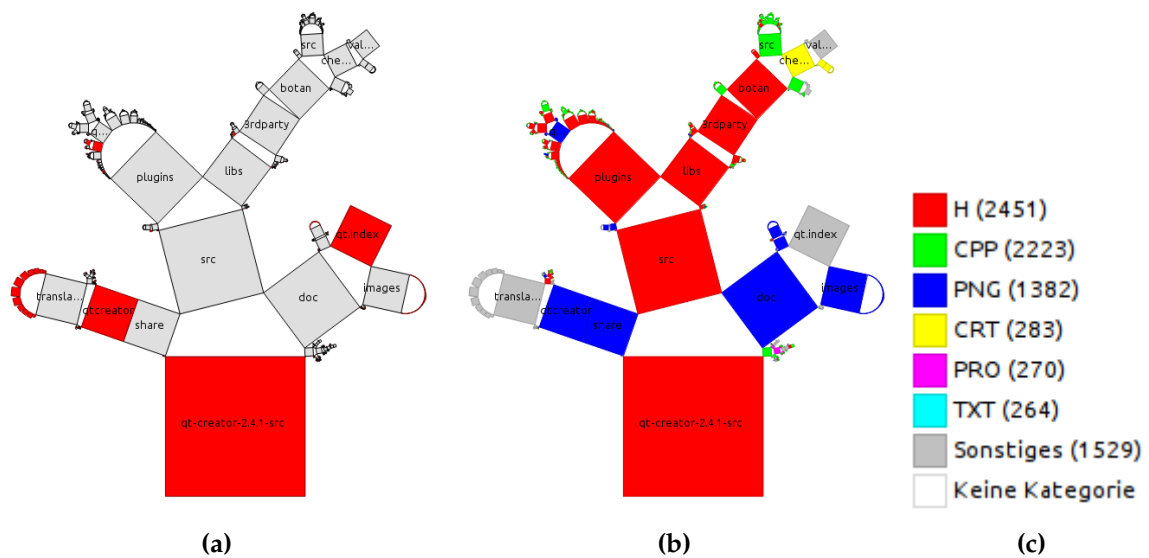


Abbildung 5.3: (a) Suchergebnis bei der Suche nach der Zeichenkette „qt“. (b) Häufigkeitsverteilung für Dateieindungen: die Knoten sind nach Häufigkeit von Dateieindungen im Unterbaum eingefärbt und (c) die zugehörige Legende mit Informationen über die Anzahl der Vorkommnisse der einzelnen Dateiartern.

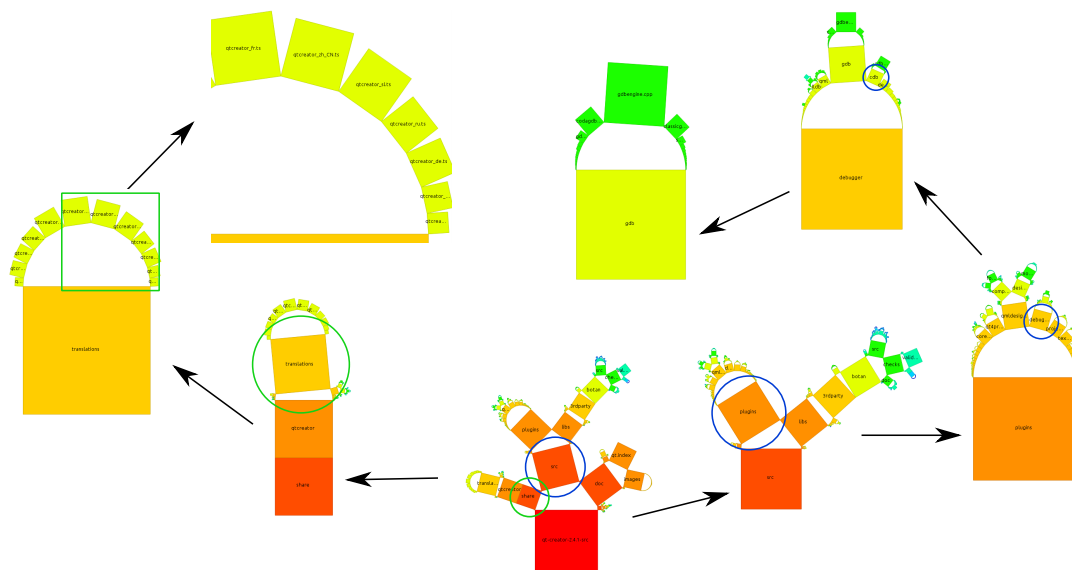


Abbildung 5.4: Veranschaulichung der Navigation innerhalb der Verzeichnisstruktur. Zwei Pfade wird gefolgt. Die grün markierten Bereiche führen zu den Übersetzungsdateien des Programms; die blau markierten zum Quellcode für den Debugger. Dabei wird in jedem dargestellten Schritt entweder ein Teilbaum der Hierarchie dargestellt (Kreis) oder zu einem Bereich geometrisch gezoomt (Rechteck).

Die linke Hälfte der Darstellung zeigt einen Weg zu den Übersetzungsdateien. Wenn der Zielordner erreicht ist, können durch geometrisches Zoomen die verschiedenen Namen der Übersetzungsdateien angezeigt werden. Die rechte Hälfte zeigt den Weg zu den Debugger-Quellcode-Dateien.

5.2 Phylogenetischer Baum (NCBI-Taxonomie)

Als nächstes wird ein phylogenetischer Baum betrachtet. Hierzu wird die NCBI-Taxonomie [Fed02] verwendet, die vom National Center for Biotechnology Information entwickelt wurde und die wissenschaftlichen Namen und Klassifikationen aller Organismen enthält, die in deren Gen-Datenbank [BKML⁺09] enthalten sind. Sie veranschaulicht die evolutionären Beziehungen zwischen Organismen und wird häufig in der Biologie und Bioinformatik verwendet.

Diese Hierarchie liegt im Newick-Format (ohne Größeninformationen) vor. Sie besteht aus insgesamt 324276 Knoten (60585 Klassen und 263691 Arten) und hat eine maximale Tiefe von 42. Hierbei wird jede Klasse durch einen inneren Knoten repräsentiert und die Arten durch Blattknoten. Der höchste Verzweigungsgrad eines Knotens beträgt 6513. Der Wurzelknoten ist ein künstlicher Knoten, der jede Spezies enthält, für die eine DNA-Sequenz oder ein Protein im digitalen NCBI-Archiv gespeichert ist.

In Abbildung 5.5 sind verschiedene Darstellungsformen der Hierarchie zusammen mit dem verwendeten Farbschema (Abbildung 5.5c), das in Bezug auf die Tiefe der Knoten angewendet wird, abgebildet. In Abbildung 5.5a hat jeder Knoten der gleichen Tiefe eine gleich lange Grundlinie. Während hier lediglich eine Übersicht über die Verzweigungsstruktur geliefert wird, berücksichtigt die Visualisierung in Abbildung 5.5b auch die Anzahl der Blattknoten. Dies hat zur Folge, dass klar erkennbar ist, in welchem Unterbaum die meisten Blattknoten vorhanden sind. Obwohl der Datensatz eine sehr komplexe und große hierarchische Struktur besitzt, liefert diese Visualisierung einen guten Überblick über die Struktur. Ein Hauptpfad ist deutlich zu erkennen, bestehend aus den Klassen, die die meisten Spezies enthalten.

Ein Ziel, das bei diesem Datensatz verfolgt werden könnte, ist das Finden einer bestimmten Spezies und Erforschen deren Abstammung und Nachbarschaft. Nachfolgend wird beispielhaft die Abstammung von Menschen (*Homo Sapiens*) analysiert.

Eine Suche nach „*Homo_sapiens*“ liefert zwar das Ergebnis, dass der Suchbegriff drei Mal in der Hierarchie vorkommt, allerdings sind die rot eingefärbten Knoten zu klein um sie zu finden. Wird jedoch eine Kategorieverteilung mit nur dieser einen Kategorie („*Homo_sapiens*“) erzeugt, wird aufgrund der Häufigkeitsverteilung der Pfad zu den Knoten rot eingefärbt (Abbildung 5.6). So kann entlang des Pfades zu dem gewünschten Knoten gelangt werden. Indem einzelne Knoten ausgewählt werden, ist erkennbar, an welchen Stellen andere Gattungen abzweigen oder welche Lebewesen sehr eng mit dem Menschen verwandt sind.

Durch Interaktion kann im Datensatz navigiert werden, indem gezoomt wird oder Teilbäume einzeln betrachtet werden. In Abbildung 5.7 ist der Pfad zu *Homo_sapiens* schwarz markiert.

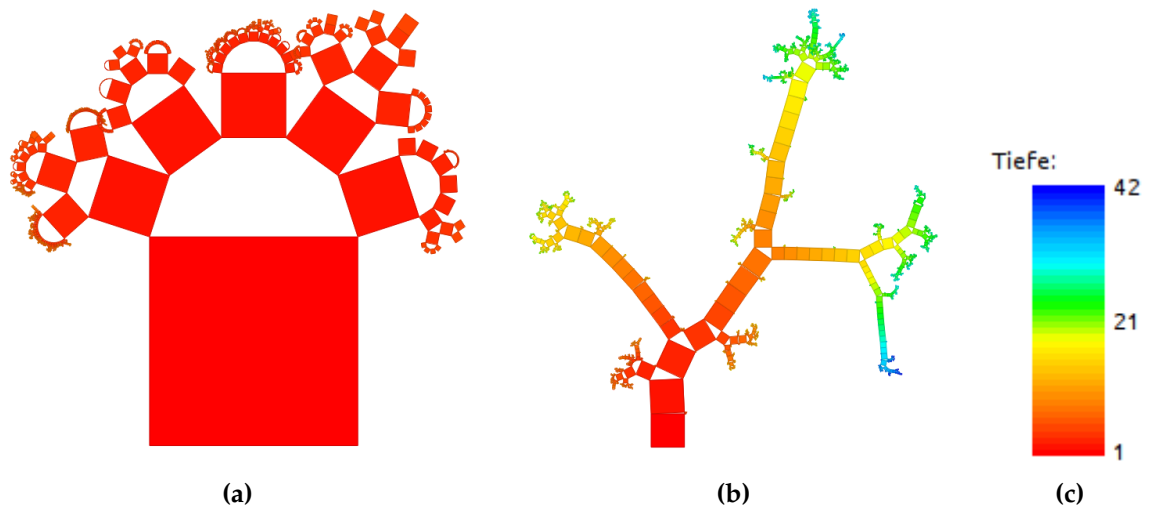


Abbildung 5.5: NCBI-Taxonomie: (a) Geschwisterknoten sind durch gleich große Quadrate dargestellt; (b) die Größe der Quadrate ist in Abhängigkeit der Anzahl der Blattknoten und (c) die verwendete Farbtabelle.

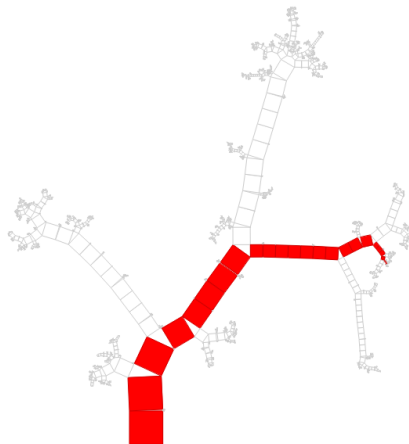


Abbildung 5.6: Der Pfad zu *Homo_sapiens* ist in der Taxonomie rot markiert.

Durch Zoomen gelangt man entlang des Pfades zu den Menschen. *Homo_sapiens* befindet sich in einer Tiefe von 30. Auf dem Pfad findet man Klassen wie Wirbeltiere, Säugetiere, Primaten, Affen, Menschenartige, Menschenaffen und dazwischen noch viele weitere.

Abbildung 5.8 zeigt die Knoten, die sich in der direkten Umgebung der Menschen befinden. Man sieht, dass der Mensch eng verwandt ist mit Schimpansen, Gorillas und Orang-Utans. Diese Arten sind innerhalb der Taxonomie nicht weit entfernt von *Homo_sapiens* und zusammen mit den Menschen als Unterkategorien der Menschenaffen zu erkennen.

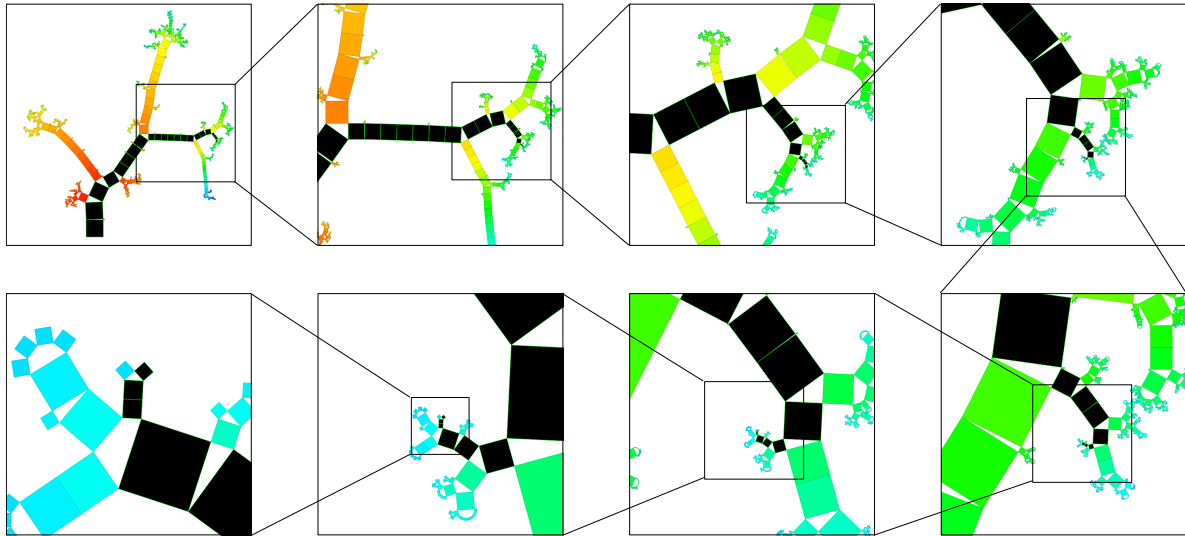


Abbildung 5.7: Geometrisches Zoomen entlang des eingefärbten Pfades zum Knoten mit der Bezeichnung *Homo_sapiens*.

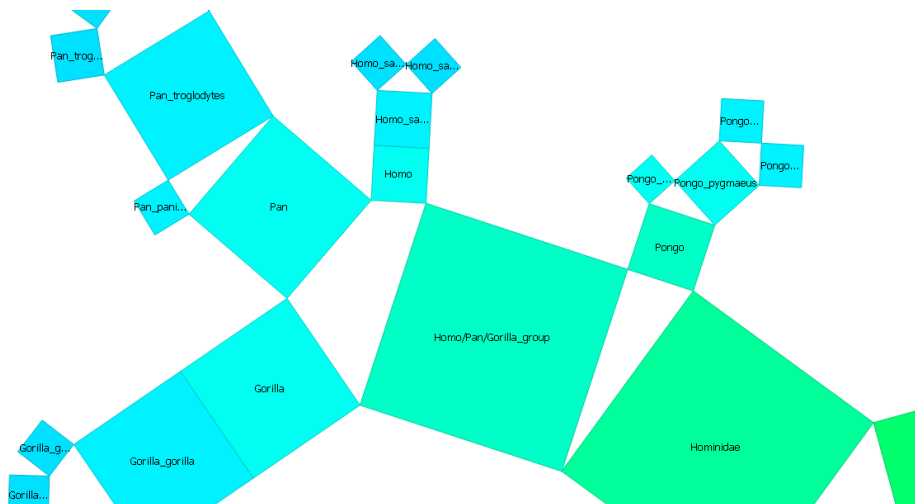


Abbildung 5.8: Teilausschnitt der NCBI-Taxonomie mit Labels. Zu erkennen sind u.a. Menschen (*Homo_sapiens*), Schimpansen (*Pan*), Gorillas (*Gorilla*) und Orang-Utans (*Pongo*).

6 Zusammenfassung und Ausblick

Ziel dieser Arbeit war die Erweiterung von Pythagorasbäumen zu verallgemeinerten Pythagorasbäumen, um beliebige hierarchische Strukturen visuell darzustellen. Außerdem sollten für die Visualisierungstechnik Interaktions- und Navigationstechniken entwickelt werden um einen besseren Einblick in einen Datensatz zu ermöglichen.

Traditionelle Pythagorasbäume wurden für Hierarchien mit beliebigem Verzweigungsgrad zu verallgemeinerten Pythagorasbäumen erweitert. Anstatt rechtwinkliger Dreiecke wurden konvexe Polygone zur Verbindung der Knoten verwendet und die Knoten durch Rechtecke dargestellt. Hierbei wurden verschiedene Parametereinstellungen für die Rechtecke implementiert und getestet. Indem Eigenschaften der einzelnen Knoten beachtet werden, wie z.B. die Anzahl von Blattknoten in einem Unterbaum oder ihre Größeninformationen, können diese durch unterschiedliche Größenverhältnisse der Rechtecke dargestellt werden. Dabei kann sowohl die Breite als auch die Höhe der Rechtecke variieren. Zusätzlich kann die Reihenfolge der Anordnung von Geschwisterknoten in unterschiedlichen Sortierungen erfolgen und eine farbliche Kodierung der Knoten, z.B. nach Tiefe, Gewicht oder Kategorie, die entsprechenden Eigenschaften zusätzlich verdeutlichen.

Es wurde festgestellt, dass die Visualisierungstechnik besonders für tiefe Hierarchien mit wenigen Verzweigungen gut geeignet ist und ästhetische Darstellungen entstehen können; ein Nachteil ist jedoch, dass teilweise Überlappungen von Knoten auftreten.

Der Ansatz wurde in einer Anwendung implementiert, in der ein Datensatz zusätzlich durch Interaktion untersucht werden kann. So ist es möglich, sowohl geometrisch als auch semantisch zu zoomen (Teilbaum anzeigen), Teile der Hierarchie können hervorgehoben werden, zusätzliche Informationen können eingeblendet werden und es kann zwischen verschiedenen Teilbäumen navigiert werden.

Anhand von zwei Datensätzen wurde gezeigt, wie die Anwendung sinnvoll eingesetzt werden kann. Als erstes wurde eine Verzeichnisstruktur untersucht, bei der verschiedene Visualisierungseinstellungen verwendet und demonstrativ Interaktionstechniken eingesetzt wurden. Danach wurde anhand eines phylogenetischen Baumes die Abstammung des Menschen betrachtet.

Ausblick

Es gibt noch weitere Möglichkeiten, die Visualisierung und die Interaktionsmöglichkeiten für die verallgemeinerten Pythagorasbäume zu verbessern und auszubauen.

Wünschenswert wäre ein Verfahren, das Überlappungen in einer Darstellung reduziert. Beispielsweise könnte automatisch die Reihenfolge der Knoten verändert oder die geometrischen Eigenschaften der einzelnen Knoten in Bezug auf die Höhe und Breite der Rechtecke angepasst werden. In der Anwendung können momentan zwar verschiedene Einstellungen verändert werden (z.B. die Knotenanordnung), jedoch ist es hier oft nicht möglich, eine Darstellung ohne Überlappungen zu finden.

Beim Ein-/Ausblenden verschiedener Teile der Visualisierung kann es für den Benutzer teilweise schwierig sein, genau nachzuvollziehen, was sich verändert hat bzw. welcher Zusammenhang zu der zuvor dargestellten Visualisierung besteht. Um einem Verlust der Mental Map vorzubeugen, könnte Animation verwendet werden. Dadurch kann die Fähigkeit, Zusammenhänge zu rekonstruieren und zu verstehen, verbessert werden [BB99]. Wenn Knoten aggregiert werden, könnte dieser Vorgang durch Animation erfolgen. Aber auch wenn ein Teilbaum ohne die restlichen Knoten dargestellt wird, könnte eine Rotation und ein Ausblenden der restlichen Hierarchie den Prozess für den Benutzer verständlicher machen. Das gleiche gilt auch beim Wiedereinblenden von weiteren Hierarchieebenen oder beim Zoomen zu bestimmten Knoten.

Die Suchfunktion ermöglicht momentan nur einen Überblick über die gefundenen Knoten durch Einfärbung der entsprechenden Knoten in der Visualisierung und durch eine Information über die Anzahl der gefundenen Ergebnisse. Zum einen sind die Knoten, wenn sie sehr klein sind, in der Visualisierung teilweise nur schlecht zu erkennen, zum anderen wären weitere Interaktionsmöglichkeiten mit den Ergebnissen sinnvoll. Eine Auflistung aller Suchergebnisse durch die der Benutzer zu den einzelnen Knoten gelangen kann, sodass sie z.B. ausgewählt werden oder zu ihnen gezoomt werden kann, wäre eine Überlegung wert.

Um die Hierarchie interaktiv zu verändern, könnte, ähnlich wie in einem Dateimanager, das Verschieben, Löschen und Kopieren von Unterbäumen erlaubt werden.

Außerdem könnte eine Benutzerstudie weitere Einblicke geben, wie sinnvoll und verständlich diese neue Visualisierungstechnik ist. Es könnten weitere Vor- und Nachteile der Visualisierung aufgedeckt werden und besondere Merkmale zum Vorschein kommen.

Literaturverzeichnis

- [BB99] B. Bederson, A. Boltman. Does Animation Help Users Build Mental Maps of Spatial Information? In *Proceedings of IEEE Symposium on Information Visualization, 1999. (Info Vis '99)*, S. 28–35. 1999. doi:10.1109/INFVIS.1999.801854. (Zitiert auf Seite 41)
- [BDLo5] M. Balzer, O. Deussen, C. Lewerentz. Voronoi Treemaps for the Visualization of Software Metrics. In *Proceedings of the 2005 ACM symposium on Software visualization, SoftVis '05*, S. 165–172. ACM, New York, NY, USA, 2005. doi:10.1145/1056018.1056041. URL <http://doi.acm.org/10.1145/1056018.1056041>. (Zitiert auf Seite 8)
- [BHP⁺12] F. Block, M. Horn, B. Phillips, J. Diamond, E. Evans, C. Shen. The DeepTree Exhibit: Visualizing the Tree of Life to Facilitate Informal Learning. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2789–2798, 2012. doi:10.1109/TVCG.2012.272. (Zitiert auf Seite 9)
- [BHW00] M. Bruls, K. Huizing, J. Wijk. Squarified Treemaps. In W. Leeuw, R. Liere, Herausgeber, *Data Visualization 2000*, Eurographics, S. 33–42. Springer Vienna, 2000. doi:10.1007/978-3-7091-6783-0_4. URL http://dx.doi.org/10.1007/978-3-7091-6783-0_4. (Zitiert auf Seite 8)
- [BKH⁺11] M. Burch, N. Konevtsova, J. Heinrich, M. Hoferlin, D. Weiskopf. Evaluation of Traditional, Orthogonal, and Radial Tree Diagrams by an Eye Tracking Study. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2440–2448, 2011. doi:10.1109/TVCG.2011.193. (Zitiert auf Seite 7)
- [BKML⁺09] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, E. W. Sayers. Genbank. *Nucleic Acids Research*, 37(Database-Issue)(1):26–31, 2009. (Zitiert auf Seite 37)
- [Bos57] A. E. Bosman. *Het wondere onderzoekingsveld der vlakke meetkunde*. Parcival, Breda, 1957. (Zitiert auf den Seiten 4, 11 und 13)
- [BRW10] M. Burch, M. Raschke, D. Weiskopf. Indented Pixel Tree Plots. In *Advances in Visual Computing*, S. 338–349. Springer, 2010. (Zitiert auf den Seiten 4 und 9)
- [CA04] S. Carpendale, A. Agarawala. PhylloTrees: Harnessing Nature’s Phyllotactic Patterns for Tree Layout. In *IEEE Symposium on Information Visualization, 2004. INFOVIS 2004*, S. p3–p3. 2004. doi:10.1109/INFVIS.2004.53. (Zitiert auf Seite 9)
- [DK96] L. Devroye, P. Kruszewski. The Botanical Beauty of Random Binary Trees. In *Graph Drawing*, S. 166–177. Springer, 1996. (Zitiert auf Seite 9)

- [Ern85] B. Ernst. *Bomen van Pythagoras, variaties van Jos de Mey*. Aramith Uitgevers, Amsterdam, 1985. (Zitiert auf Seite 4)
- [Fed02] S. Federhen. Chapter 4 - The Taxonomy Project. In J. McEntyre, J. Ostell, Herausgeber, *The NCBI Handbook*. Bethesda (MD): National Center for Biotechnology Information, 2002. URL <http://www.ncbi.nlm.nih.gov/books/NBK21100/>. (Zitiert auf Seite 37)
- [GADMo6] S. Grivet, D. Auber, J. P. Domenger, G. Melancon. Bubble Tree Drawing Algorithm. In *Computer Vision and Graphics*, Band 32 von *Computational Imaging and Vision*, S. 633–641. Springer Netherlands, 2006. doi:10.1007/1-4020-4179-9_91. URL http://dx.doi.org/10.1007/1-4020-4179-9_91. (Zitiert auf Seite 7)
- [Gra94] E. Grabska. Graphs and Designing. In *Graph Transformations in Computer Science*, S. 188–202. Springer, 1994. (Zitiert auf Seite 13)
- [JS91] B. Johnson, B. Shneiderman. Tree-Maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd conference on Visualization '91*, VIS '91, S. 284–291. IEEE Computer Society Press, Los Alamitos, CA, USA, 1991. URL <http://dl.acm.org/citation.cfm?id=949607.949654>. (Zitiert auf den Seiten 4 und 8)
- [JS10] S. Jürgensmann, H.-J. Schulz. Poster: A Visual Survey of Tree Visualization. In *Proceedings of IEEE Information Visualization*, Band 5. 2010. (Zitiert auf den Seiten 4 und 7)
- [KL83] J. B. Kruskal, J. M. Landwehr. Icicle Plots: Better Displays for Hierarchical Clustering. *The American Statistician*, 37(2):162–168, 1983. (Zitiert auf den Seiten 4 und 8)
- [Knu72] D. E. Knuth. *Art of Computer Programming Volume 1: Fundamental Algorithms*. Addison-Wesley Publishing Company, 1972. (Zitiert auf Seite 5)
- [KWW01] E. Kleiberg, H. Van de Wetering, J. J. van Wijk. Botanical Visualization of Huge Hierarchies. In *IEEE Symposium on Information Visualization, 2001. INFOVIS 2001*, S. 87–94. 2001. doi:10.1109/INFVIS.2001.963285. (Zitiert auf Seite 9)
- [KY93] H. Koike, H. Yoshihara. Fractal Approaches for Visualizing Huge Hierarchies. In *Proceedings of IEEE Symposium on Visual Languages, 1993*, S. 55–60. IEEE, 1993. (Zitiert auf Seite 9)
- [Lau89] H. A. Lauwerier. *Fraktale verstehen und selbst programmieren*. Wittig-Fachbuchverl., Hückelhoven, 1989. (Zitiert auf Seite 13)
- [LY07] C.-C. Lin, H.-C. Yen. On Balloon Drawings of Rooted Trees. *Journal of graph algorithms and applications*, 11(2):431–452, 2007. (Zitiert auf Seite 7)
- [Man83] B. B. Mandelbrot. *The Fractal Geometry of Nature*. W.H. Freeman and Company, New York, 1983. (Zitiert auf den Seiten 4 und 9)

- [Mey10] J. de Mey. *Illusionistische Malerei: Bilder aus privaten Sammlungen*. Edition Xim Virgines, 1 Auflage, 2010. (Zitiert auf Seite 4)
- [OLY05] T. Ong, J. J. Leggett, U. Yun. Visualizing Hierarchies and Collection Structures with Fractal Trees. In *Sixth Mexican International Conference on Computer Science, 2005. ENC 2005*, S. 31–40. 2005. doi:10.1109/ENC.2005.53. (Zitiert auf Seite 9)
- [PJS92] H.-O. Peitgen, H. Jürgens, D. Saupe. *Introduction to fractals and chaos*. Fractals for the classrooms. Springer-Verlag, 1992. (Zitiert auf Seite 13)
- [qtc12] qt-creator-2.4.1-src.zip, 2012. URL <http://download.qt-project.org/archive/qtcreator/2.4/qt-creator-2.4.1-src.zip>. (Zitiert auf Seite 32)
- [qtp] Qt Project. URL <http://qt-project.org/>. (Zitiert auf Seite 32)
- [RH12] J. Rosindell, L. J. Harmon. OneZoom: A Fractal Explorer for the Tree of Life. *PLoS Biol*, 10(10):e1001406, 2012. doi:10.1371/journal.pbio.1001406. URL <http://dx.doi.org/10.1371/journal.pbio.1001406>. (Zitiert auf Seite 9)
- [Sch11] H.-J. Schulz. Treevis.net: A Tree Visualization Reference. *Computer Graphics and Applications, IEEE*, 31(6):11–15, 2011. (Zitiert auf Seite 4)
- [Spe07] R. Spence. *Information visualization*. Pearson, London u.a., 2007. (Zitiert auf Seite 28)
- [SZ00] J. Stasko, E. Zhang. Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations. In *IEEE Symposium on Information Visualization, 2000. InfoVis 2000*, S. 57–65. IEEE, 2000. (Zitiert auf Seite 8)
- [Wet03] K. Wetzel. Pebbles – using Circular Treemaps to visualize disk usage. <http://lip.sourceforge.net/ctreemap.html>, 2003. (Zitiert auf Seite 8)
- [YWR02] J. Yang, M. O. Ward, E. A. Rundensteiner. InterRing: An Interactive Tool for Visually Navigating and Manipulating Hierarchical Structures. In *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002*, S. 77–84. IEEE, 2002. (Zitiert auf Seite 8)

Alle URLs wurden zuletzt am 26.05.2013 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift