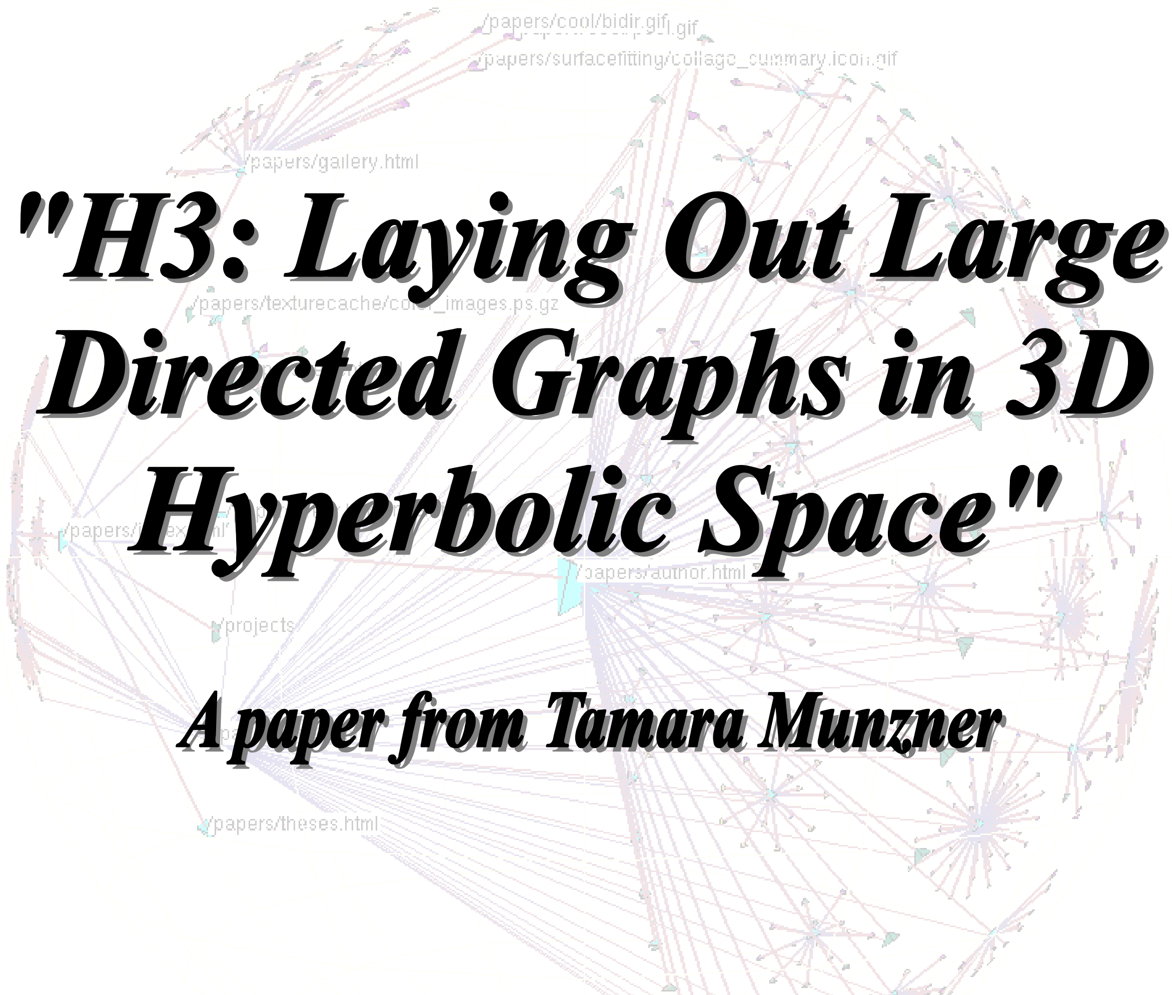
Eidgenössische Technische Hochschule Zürich
Computer Graphics Research Group

Fachseminar
SS1999



"H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space"

A paper from Tamara Munzner

Pascal Glardon
Abteilung für Informatik

1 Einleitung

H3 ist eine Methode für die Darstellung grosser gerichteter Graphen, wie eine Web Struktur, im drei dimensionalen Raum. Diese Methode ist eine Erweiterung der schon existierenden Methoden für solche Darstellungen. H3 hat mehrere Vorteile verglichen mit den bisherigen Ideen.

- Dank der Benutzung der natürlichen Hierarchie der gegebenen Daten können wir grössere Strukturen darstellen.
- Durch die Benützung des hyperbolischen statt des euklidischen Raumes, verfügt man über einen unendlichen Raum für unsere Daten.
- Es entsteht eine Verbesserung der traditionellen Darstellung im 3D hyperbolischen Raum. Anstatt die Kinder auf dem Umkreis der Basis eines Kegels einzusetzen, legen wir eine Hemisphäre auf die Öffnung des Kegels und setzen dann alle Kinder auf die Oberfläche der Kalotte.

2 Kurze Beschreibung des H3

Wie der Titel dieser Arbeit schon erwähnt, ist H3 eine Methode, ein Algorithmus für die Darstellung **gerichteter Graphen** ("directed graphs"). Diese Graphen bilden eigentlich eine natürliche Repräsentation einer Struktur. Sie sind weit verbreitet. Im Bereich "Information Systeme" werden sie sehr oft verwendet. Viele Strukturen, die mit der Informatik zu tun haben, lassen sich als Graphen mit Knoten und Pfeilen bezeichnen. Ein berühmtes Beispiel ist die sogenannte EBNF Notation. Wir können auch an die Dateistruktur einer Harddisk denken. Aber auch im Bereich der Studien können wir diese Darstellung verwenden, zum Beispiel für die heuristischen Schemas.

Das Problem in einem Graph ist, dass wir noch kein **Spanning Tree** haben. Man muss also einen solchen konstruieren, um eine wirksame Darstellung zu erreichen. Zwei Methoden sind vorhanden.
Die einfachste Lösung: das Herstellen eines Breadth-first Search. Leider stellt die resultierende Darstellung einen groben und schlechten Eindruck der Struktur vor.
Die bessere Lösung: die Informationen von Knoten benützen, um eine Klassifikation zu erzeugen. Das heisst, mit den **"Kenntnissen" der Knoten** können wir eine hierarchische Struktur und einen Spanning Tree aufbauen, der wahrscheinlich ein besseres mentales Modell bietet. Mehr über dieses Thema befindet sich im Kapitel 4.1.
Die zweite Variante wird in H3 benützt mit einer Behandlung der **Nontree Links**. Die Nontree Links sind die Kanten, welche nicht im Spanning Tree vorkommen. Wir können also wählen, ob wir diese Kanten darstellen wollen oder nicht.

H3 stellt die Daten in einem **drei dimensionalem Raum** dar. Jetzt kommt die Frage, welche Geometrie benützt wird. Als erstes springt einem die euklidische Geometrie ins Auge. Beim genaueren betrachten erkennt man die zweite Möglichkeit, das ganze im hyperbolischen Raum zu modellieren.
Der euklidische Raum ist die einfachere Lösung. Wir sind gewöhnt in diesem Raum zu rechnen. Aber wenn wir ein bisschen weiter denken, so gibt es Probleme mit dem zur Verfügung stehenden Platz. Tatsächlich wächst eine Baumstruktur exponentiell, aber der Umkreis oder die Fläche im euklidischen Raum wächst nur polynomiell. Das heisst, wir müssen weniger Raum für die Knoten, die tief in der Struktur sind, allozieren. Und wenn wir eine allgemeine Ansicht der Struktur sehen wollen, werden nur die benachbarten Knoten von

der Wurzel unserer Struktur dargestellt. Leider können wir keine tiefere Stufe sehen. Dafür müssen wir eine Vergrösserung ausführen. Wir haben also keine allgemeine Übersicht der Struktur.
Die beste Lösung ist die Benützung des **hyperbolischen Raumes**. Dieser Raum hat den Vorteil, dass er auch exponentiell wächst (wie die Anzahl der Knoten). Wir können also den gleichen Raum für jeden Knoten allozieren, unabhängig wie tief die Knoten sind. Dann wird eine Projektion in einen Rauminhalt des euklidischen Raumes gemacht, zum Beispiel eine Darstellung am Bildschirm. Mit diesem hyperbolischen Raum verlieren wir die allgemeine Ansicht der Struktur nicht, was beim euklidischen Raum der Fall war. Eine genauere Erklärung über diese Lösung wird im Kapitel 5 vorgestellt.

Für ein besseres Verständnis des H3 Algorithmus wird ein traditionelles Beispiel erwähnt, nämlich die **Struktur eines Website**. Es gibt tatsächlich viele Anwendungen für eine solche Darstellung.
Der Surfer kann sich in einem Site lokalisieren. Er hat eine gut verarbeitete Karte des Ortes, besser als eine einfache Liste der möglichen Links. Möglich ist auch die ganze Reihe der schon besuchten Seiten darzustellen.
Ein Webmaster kann mit Hilfe dieser Darstellung seine eigene statische Struktur aber auch die dynamische Struktur anschauen, das heisst der Verkehr auf seinen verschiedenen Seiten.
Die Ergebnisse einer Suche auf dem Internet, können auch auf diese Weise dargestellt werden, indem die Wichtigkeit als Hierarchie betrachtet wird.

3 Die Vorgänger des H3

H3 wurde 1997 an der InfoVis präsentiert. Der freie Software **Site Manager**, der für die Verwaltung eines WebSite hilft (http://www.sgi.com/software/sitemgr.html), braucht diesen H3 Algorithmus. Aber es gibt eine Reihe von Programmen, welche mit der Darstellung von Daten zu tun haben. Die Entwicklung solcher Programme hat in den neunziger Jahren angefangen. Sowohl im zwei dimensionalen, als auch im drei dimensionalen Fall. Die Focus & Context Technik wird am meistens im hyperbolischen Raum benützt.

3.1 2D Graph und Tree

In zwei Dimensionen gibt es mehrere Systeme für die Web Darstellung. **Webmap** war eines der ersten Systeme, dann das **WebViz** System mit einer zufälligen Darstellung der Knoten. Das MosaicG System macht einen geschichtlichen Verlauf in zwei Dimensionen. MagniFind von Inxight kann das Filesystem von einem PC im hyperbolischen Raum darstellen.
Diese Systeme sind gut geeignet für eine kleine Menge von Knoten. Aber wenn die Anzahl der Knoten Tausend übersteigt, funktioniert ein solches System nicht mehr. Wir gehen also in eine höhere Dimension über.

3.2 3D Graph und Tree

Für die Darstellung in drei Dimensionen unterscheidet man die Graph Struktur und die Tree Struktur.

Im Fall eines **Graphen** gibt es zwei mögliche Modelle für die Darstellung: Ebene Schichten oder Federmasse System.

Für das erste Modell bietet die Firma SGI ein 2½ dimensionales Filesystem: **fsn**. Wir setzen also ein File in eine bestimmte Ebene ein. Wir haben also zwei Freiheitsgrade für die Plazierung in einer Ebene und einen halben Freiheitsgrad für die Wahl einer Ebene in eine vorgeschlagene Liste. Das **Harmony Information Landscape** nützt den drei dimensionalen Raum aus. Die Links sind über oder unter einer bestimmten Ebene.
Das zweite Modell basiert auf das Federmasse System. Die Knoten sind als Masse bezeichnet und stossen sich gegenseitig ab. Die Links sind als Feder bezeichnet und üben eine anziehende Kraft aus. Wir können also ein Differentialgleichungssystem bearbeiten und es in einer iterativen Form auflösen. Das klappt wenn die Anzahl der Knoten nicht grösser als ein paar Tausend ist. Zwei Systeme benützen dieses Modell: **Gem3D** und **Hyper/Narcissus**. Dieses letztere baut einen Graph mit dem semantischen Inhalt der Knoten auf. Es ist zu beachten, dass schon eine Graphstruktur vorgegeben sein muss.

Im Fall eines **Trees** wird ein spezielles Darstellungsmodell benützt: das "Cone Tree" System. Eine genauere Beschreibung dieses Modells steht im Kapitel 4.2. **Xerox PARC** hat diese Technik als erstes eingeführt. Viele Systeme wurden von diesem Modell abgeleitet. Die neue Version von **WebViz** hat noch einen Schritt weiter gemacht: der Cone Tree wird im hyperbolischen Raum eingetragen.
Betreffend das H3 System, handelt es sich um eine Mischung von einer Darstellung des Graphen in drei Dimensionen, mit einer erweiterten "Cone Tree" Methode.

3.3 Die Focus & Context Technik

Focus & Context ist eine Technik, die eine grosse Menge von Daten in einer bestimmten Bildschirmfläche erlaubt, indem eine gewisse Distorsion eingefügt wird. Um diese Technik zu erreichen, empfehlen die Verteidiger der "Cone Tree" Methode, eine euklidische Perspektive. Eine andere Betrachtungsweise besteht darin, den Graphen durch eine **Fischauge Linse** zu sehen. Weiter kann der Graph auch einem dehnbaren Streifen ausgelegt werden. Der **2D hyperbolic tree browser** von Xerox PARC entsteht aus der ersten Überlegung im hyperbolischen Raum. Das **WebViz** System benützt auch den hyperbolischen Raum, aber das Verhältnis zwischen den gesamten Raum und der Menge der Daten ist zu gross.
Die verwendete Methode in H3 ist ähnlich wie die Fischauge Linse im hyperbolischen Raum.

Die Tabelle 1 fasst alle bisherige Methoden der Darstellung von Daten, sowohl in zwei als auch in drei Dimensionen, im euklidischen oder im hyoperbolischen Fall zusammen.

4 Layout

Der H3 Algorithmus für die Darstellung setzt sich aus zwei Phasen zusammen. Erstens müssen wir ein Spanning Tree von einem Input Graph aufbauen und zweitens müssen wir die Stelle für jedes Element in unserem Raum bestimmen. Danach können wir die Struktur darstellen und durch die Struktur navigieren.

4.1 Vom Graph zum Spanning Tree

Die Bildung des Spanning Tree ist sehr wichtig. Wir wollen einen guten visuellen Eindruck der Struktur. Zwei Methoden sind vorhanden: eine einfache, die schon die existierende Struktur bearbeitet und eine geeignetere, die die Kenntnisse der Daten bearbeitet.

Diese zweite Variante läuft folgendermassen. Von einer Menge ungeordneter Daten wählen wir einige Ausdrücke, von welchen jeder eine Klasse definiert. Ein Beispiel wäre eine Sammlung von Bilder-, Text- und Soundfiles. Dann wird ein Spanning Tree mit den Kriterien Bild, Text und Ton aufgebaut. Dieser gebildete Spanning Tree ist den Erwartung des Benützers sehr nah.

<table>
<tr><td colspan="3"></td><td>Euklidischer Raum</td><td>Hyperbolischer Raum</td></tr>
<tr><td>2D</td><td colspan="2">Graph & Tree</td><td>WebViz (I)
MosaicG</td><td>Hyperbolic Tree
Browser (Xerox)
Inxight</td></tr>
<tr><td rowspan="3">3D</td><td rowspan="2">Graph</td><td>Ebene</td><td>SGI fsn (2½ D)
HIL</td><td></td></tr>
<tr><td>Feder-Masse</td><td>Gem3D
Hyper/Narcissus</td><td></td></tr>
<tr><td>Tree</td><td>Cone Tree</td><td>Xerox PARC</td><td>Persp. Projektion
Fischauge Linse
Ausdehnbare Streifen
WebViz (II)
H3</td></tr>
</table>

Tabelle 1

Um dieses zu erklären, nehmen wir zum Beispiel, die Hyperlink Struktur von einer Webseite. Die Knoten stehen für die Web Dokumente und Kanten, oder besser Pfeile, stehen für die Hyperlinks zwischen den Dokumenten. Die Erzeugung des Spanning Trees sieht folgendermassen aus: das Verfahren sucht die Datei, welche mit *index. html* bezeichnet wird. Diese Datei wird als Eltern betrachtet und die Kinder sind alle Dateien, welche von diesem index-file direkt referenziert werden. Ein Vorteil dieser Methode ist, dass wir eine präzise visuelle Analyse der Struktur machen können. Die verwaisten Dokumente sind einfach sichtbar und ermöglichen eine Korrektur.

4.2 Tree Layout

H3 Algorithmus beruht auf einer erweiterten Cone Tree Methode. Die ursprüngliche Methode besteht aus Kinder-Knoten auf dem Umfang eines Kegelöffnung. Der Vater liegt auf der Spitze des Kegels. Diese Lösung ist eigentlich nicht gut geeignet, wenn die Anzahl der Knoten zu gross ist. Eine Überfüllung wird sich ergeben. H3 setzt die Kinder auch nicht auf dem Umfang ein, sondern auf der **Oberfläche einer Hemisphäre**, die im Kegelmund plaziert wird. Das Bild 1 zeigt den Gewinn von Platz mit dieser Methode.

Um diese Lösung zu realisieren brauchen wir den Radius der Kalotte – das wird in der **bottom-up** Phase berechnet – und die Stelle jedes Knotes auf dieser Kalotte – das wird in der

top-down Phase berechnet. Es ist noch zu beachten, dass diese zwei Phasen nicht parallel ausgeführt werden können, weil die zweite Phase die berechneten Radien in der ersten Phase braucht.

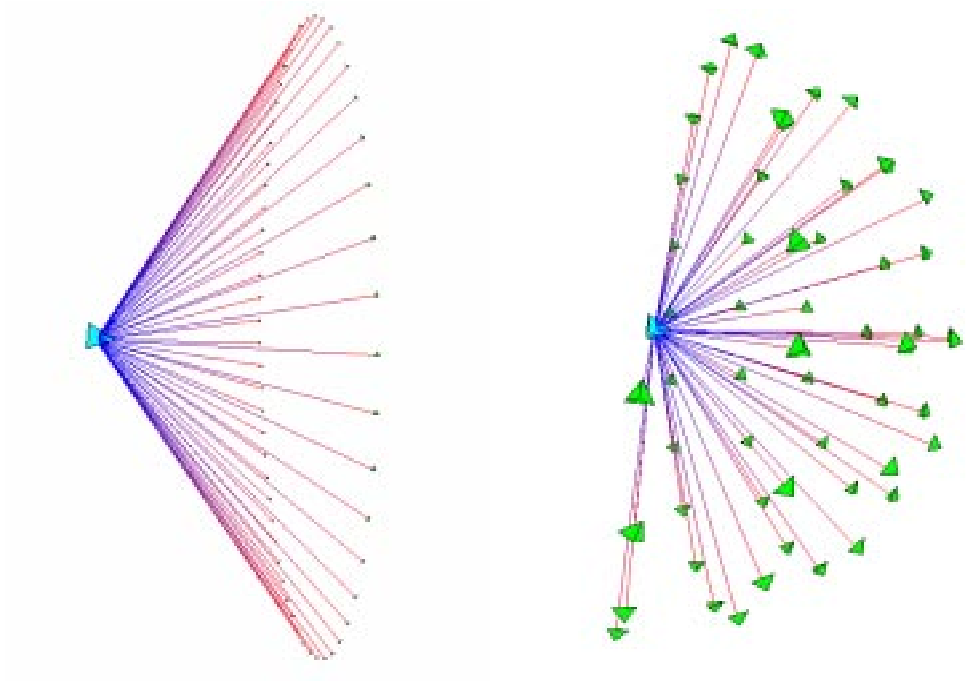


Bild 1: links Knoten auf dem Umfang und rechts auf der Hemisphäre

4.2.1 Bottom-up Phase

Die bottom-up Phase beginnt an der Stufe der Blätter (bottom) und endet an die Wurzel des Graphen (up). Ein Schritt dieser Phase könnte folgendermassen durchgeführt werden. Wir summieren alle Flächen, die von den zugehörigen Kindern auf der Kalotte bedeckt werden und erhalten die Fläche der gesuchten Hemisphäre. Aber zur Berechnung der von den Kindern bedeckten Fläche, benützten wir den Radius der Kalotte des Vaters, was eigentlich gesucht ist! Also dieses Verfahren geht nicht. Eine andere und gute Lösung ist, anstatt jede bedeckte Fläche zu summieren, den Deckel von jeder Kinderkalotte zu nehmen. Der Radius jedes Knotens kann einfach durch eine Traversierung seiner Nachkommen berechnet werden. Dann berechnen wir die Fläche dieser Scheibe. Im Fall eines Blattes weisen wir mehr Raum zu als nötig, um eine Überfüllung zu vermeiden. Diese Approximation funktioniert bei kleinen

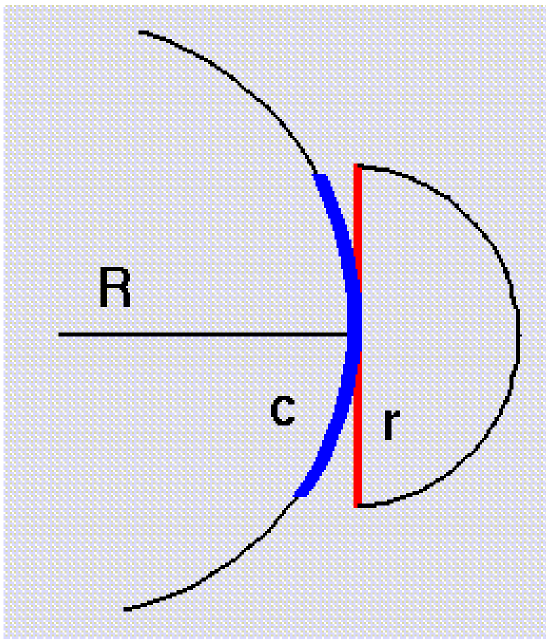


Bild 2

Winkeln. Die Abbildung 2 zeigt in Blau die richtige Lösung und in Rot die gewählte Approximation.

Nach diesen Überlegungen sind wir bereit eine mathematische Herleitung zu bilden. Da wir im hyperbolischen Raum arbeiten, müssen wir auch mit hyperbolischen Formeln rechnen. Die Tabelle 2 zeigt genauer, wie die Übereinstimmung von der euklidischen und der hyperbolischen Geometrie funktioniert. Wir berechnen die Summe der Kinderdeckel DS mit eigenem Radius r_k, welche der Oberfläche der Vaterhemisphäre vom Radius R gleich sein muss.

$$DS = \sum_{k=1}^{n} 2\pi(\cosh(r_k)-1) = 2\pi\sinh^2(R)$$

Es bleibt noch den Radius zu isolieren:

$$R = \sinh^{-1}\left(\sqrt{\frac{DS}{2\pi}}\right)$$

Wir haben jetzt die erste Phase durchgeführt.

Euclidean and Hyperbolic Formulas		
Formula	Euclidean	Hyperbolic
right-angle triangle	$\tan\theta = \frac{opp}{adj}$	$\tan\theta = \frac{\tanh(opp)}{\sinh(adj)}$
right-angle triangle	$\sin\theta = \frac{opp}{hyp}$	$\sin\theta = \frac{\sinh(opp)}{\sinh(hyp)}$
circle area	πr^2	$2\pi(\cosh(r) - 1)$
hemisphere area	$2\pi r^2$	$2\pi\sinh^2(r)$
spherical cap area	$2\pi r(1 - \cos\phi)$	$2\pi\sinh r(1 - \cos\phi)$

Tabelle 2

4.2.2 Top-down Phase

Die top-down Phase beginnt an die Wurzel (top) und endet an die Blätter (down). Ein Schritt dieser Phase bestimmt die Stelle der Kinder auf der Kalotte eines Vaters. Dieses Problem gehört zum **sphere-packing Problem** der Familie. In unserem Fall handelt es sich um eine Kalotte (nicht wie allgemein eine Kugel) und die zu verteilenden Knoten haben nicht die gleiche Grösse.

Wir müssen einen gewissen Platz, der als Kreis bezeichnet wird, für jedes Kind auf der Kalotte des Vaters allozieren. Die Lösung dieses Problems muss schneller als präzis sein. Das ist klar, da wir nur eine generelle Ansicht über die Struktur erhalten wollen. Zuerst sortieren wir alle Kreise nach einer aufsteigenden Reihenfolge, dann plazieren wir einen nach dem anderen rings um den Äquator. Wir beginnen mit den grössten Kreisen am Pol und je kleiner diese Kreise sind, desto näher am Äquator befinden sie sich. Demzufolge rollen wir Streifen

verschiedener Breiten auf die Kalotte auf, wobei die schmalen am nächsten beim Äquator liegen.
Diese Methode ist aus zwei Gründen nicht die optimale. Erstens können wir beim schlimmsten Fall nur mit einem Kreis die ersten Streifen füllen. Zweites müssen wir eine Approximation der durch die Kinderkalotten bedeckten Fläche auf der Kalotte machen. Wir nehmen an, dass jene Fläche einer Scheibe entspricht. Um diese Approximation zu verbessern, multiplizieren wir den Radius mit einem gewissen Faktor.
Zu bemerken ist, dass die optimale Lösung immer Raum zwischen den Kreisen verliert. Die Abbildung 3 zeigt die Verteilung der Kreise auf der Kalotte.

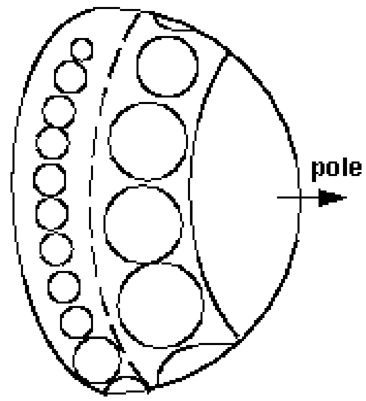


Bild 3

Die mathematische Herleitung läuft so. Wir müssen für jeden Kreis die genaue Position berechnen. Wir brauchen die sphärischen Koordinaten (R, θ_k, ϕ_k) für jedes Kind k. Der Radius R wurde schon in der bottom-up Phase berechnet. Es bleiben noch die zwei Winkel zu bestimmen.
Zuerst wollen wir mit dem θ_k-Winkel die aktuelle geographische Länge des Zentrums der betrachteten Kreises auf einer bestimmten Breite bestimmen. Wenden wir den Sinus im euklidischen Raum an:

$$r_\phi = R\sin(\phi)$$

Übertragen in den hyperbolischen Raum erhält man:

$$r_\phi = \sinh^{-1}(\sinh(R)\sinh(\phi))$$

Mit der Tangens finden wir den $\delta\theta_k$-Winkel, der die halbe Öffnung des Kreises auf der Breite bezeichnet.

$$\delta\theta_k = \arctan\left(\frac{\tanh(r_k)}{\sinh(R)\sin(\phi)}\right)$$

Wir müssen diese halbe Öffnung der Kreises k mit dieser der Kreises k-1 addieren, und das Resultat mit dem θ_{k-1}-Winkel noch summieren, was ergibt uns den θ_k-Winkel. Mit diesem Winkel können wir kontrollieren, ob der betrachtete Kreis noch Platz auf dem aktuellen Streifen hat. Wenn die Summe θ_k-Winkel und $\delta\theta_k$-Winkel grösser als 2π ist, dann gehen wir auf den nächsten Streifen und setzen wir den θ_k-Winkel zu $\delta\theta_k$.

$$\delta\phi_j = \arctan\left(\frac{\tanh(\mathrm{r_j})}{\sinh(\mathrm{R})}\right)$$

Dann bleibt noch der ϕ_j-Winkel zu bestimmen. Dieser Winkel bestimmt die aktuelle geographische Breite der betrachteten Streifen j. Wir berechnen zuerst den $\delta\phi_j$-Winkel, welcher die halbe Öffnung des grössten Kreises der Streifen j auf des Längengrades angibt.
Der Radius r_j ist der grösste Radius des Streifens j, nämlich der erste eingefügte Kreis. Wenn wir also einen Streifen weiter gehen, addieren wir diesen $\delta\phi_j$-Winkel mit demjenigen des Streifens (j-1) und dem Winkel ϕ_{j-1}, und erhalten damit die geometrische Breite des Streifens j, nämlich den Winkel ϕ_j. Die Abbildung 4 zeigt die Situation in 3D und die zwei Projektionen.

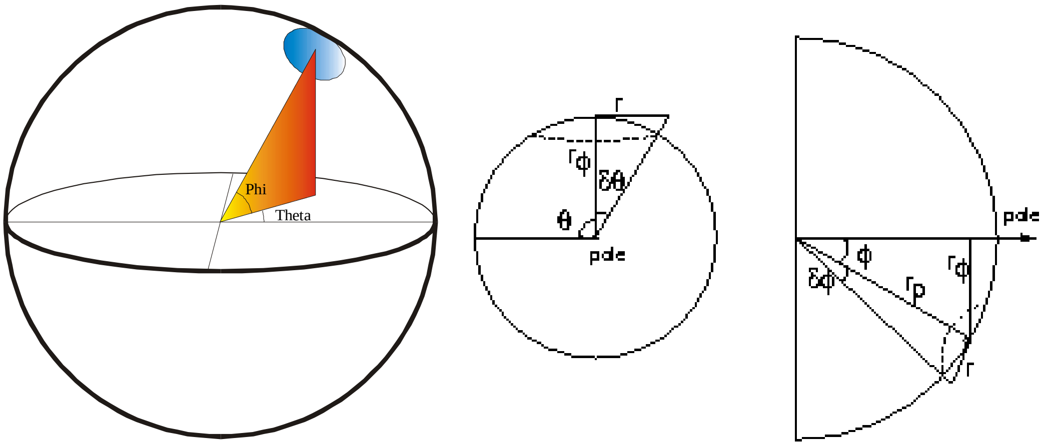


Bild 4: links die beiden Winkel, mitte Grundriss und rechts Aufriss

5 Hyperbolischer Raum

H3 ist im hyperbolischen Raum berechnet. Diese hat den erstaunlichen Vorteil, dass mehr Raum zu Verfügung steht, als im euklidischen Raum. Zum Beispiel haben im euklidischen Raum zwei parallele Linien immer den gleichen Abstand, was im hyperbolischen Raum nicht gilt. Wir können also zwei Parallele konstruieren, wo sich der Abstand immer vergrössert je weiter wir vom Ursprung entfernt sind.
Für ein unbewegliches Bild ist eine Projektion vom hyperbolischen Raum in den euklidischen Raum wie eine Projektion von einer euklidischen Szene durch eine Fischauge Linse. Jedoch ist es bei der Bewegung ganz unterschiedlich. Eine einfache Lösung wäre, euklidische Objekte im hyperbolischen Raum einzufügen und dann die Eigenschaften der hyperbolischen Geometrie anzuwenden. Aber diese Lösung nützt den exponentiellen Raum, der zu Verfügung steht, nicht.

5.1 Projektion

Es gibt zwei Projektionsverfahren vom hyperbolischen in den euklidsichen Raum, um die Darstellung zu ermöglichen.
Das **projective Modell**, welches gerade Linien erhält, aber die Winkel verzerrt, und das **conformal ball** Modell, das die Winkel erhält aber die Linien verbiegt.
In der Implementation des H3 wird das projective Modell benützt. Im eindimensionalen Fall sind die Objekte einer Hyperbel auf einem Linien Segment projeziert. Objekte, welche in der Nähe des Pols sind, sind fast nicht verzerrt. Das Bild 5 zeigt den eindimensionalen Fall. Im zweidimensionalen Fall sind Objekte auf eine Scheibe und im drei dimensionalen Fall auf eine Kugel projeziert. Wenn wir die gleichen Eigenschaften wie im eindimensionalen Fall anwenden, bemerken wir dass die Objekte, welche nicht im Zentrum liegen, verzerrt sind.

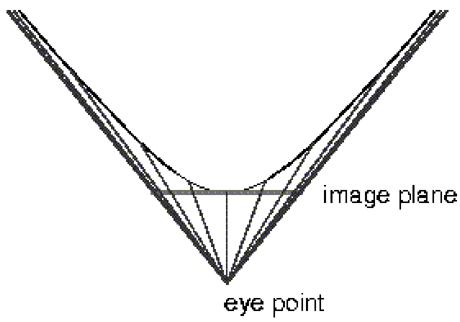


Bild 5

5.2 Präzision

Falls wir unseren Graphen auf dem Bildschirm darstellen wollen, müssen wir, wie bereits oben erwähnt, eine Projektion vom hyperbolischen in den euklidischen Raum machen. Die gebrauchte Zeit für diese Operation spielt eine Rolle für die Grösse der darstellbaren Struktur. In H3 wird es so gemacht: die hyperbolische Koordinaten, die noch unterschiedlich aber zu weit weg vom Zentrum der Kugel sind, werden fast an den Rand gesetzt. Wir können also die beiden euklidischen Koordinaten nicht mehr unterscheiden.
Knoten, welche weit vom Ursprung weg und kleiner als ein Pixel sind, brauchen nicht dargestellt zu werden. Nur bei der Bewegung der Szene können wir diese Knoten sehen.

6 Implementation

Der H3 Algorithmus wird in SiteMgr implementiert. SiteMgr ist eine Applikation von Silicon Graphics, die für die Erschaffung und die Verwaltung einer WebSite hilft. Er enthält eine interaktive Navigation der gegebenen Struktur. Wir können optional die NonTree Links darstellen. Wenn der Benützer auf einen Knoten mit der Maus klickt, kommt der Knoten durch eine Rotation und Translation ins Zentrum der Sphäre.

7 Analyse und mögliche Erweiterungen

Die H3 Technik erlaubt ohne Schwierigkeiten mehr als 20‘000 Knoten zu behandeln. Es ist also eine wirksame Lösung für die Darstellung von Nachbarschaften eines grossen Graphen in einer kleinen Bildschirmfläche.
H3 macht eine **Unterscheidung von Knoten in drei Klassen**. Die erste Klasse sind die Hauptknoten. Ihre Namen werden angezeigt. Es können 50 solche Knoten dargestellt werden. Die zweite Klasse entspricht den Knoten, die klein aber noch unterscheidbar sind. H3 kann ungefähr 500 darstellen. Dann kommt die letzte Klasse, die als Randknoten bezeichnet wird. Diese Knoten sind nicht unterscheidbar, aber sie geben trotzdem einen Eindruck auf die gesamte Struktur. 1‘000 von diesen Knoten sind sichtbar.
Ein Vorteil der Visualization in drei Dimensionen ist die **NonTree Links** zu sehen. In einer zwei dimensionalen Darstellung werden diese Links auf dem Spanning Tree gezeichnet, was beim grossen Graphen zu einer zu dichten Struktur führt. Das Problem besteht in drei Dimensionen nicht mehr. In H3 können wir die Anzeige der Nontree Links ausschalten. Das hilft wenn die Struktur sehr dicht ist.
Ein anderer Vorteil ist die Benützung des **hyperbolischen Raumes**. Tatsächlich können wir eine gute globale Ansicht der Struktur erreichen, obwohl die Anzahl der Knoten bedeutend ist.
Es ist ein Nachteil, dass die Distanzen im hyperbolischen Raum verzerrt sind. Das heisst, für einen genaueren Vergleich zwischen Knoten hilft diese Darstellung nicht viel. Ein Beispiel wäre im Bereich der Stammbäume.
H3 ist also ein sehr wirksames System, aber nicht die Lösung. Es gibt uns einen sehr wichtigen Eindruck über eine gegebene Struktur. Aber wenn die Struktur sehr dicht ist, wird es schwierig noch etwas zu erkennen. Manche Operationen, wie ein Element von einer linearen Liste zu selektieren, sind leichter in einem traditionellen zwei dimensionalen Browser auszuführen. H3 ist also ein ergänzendes Werkzeug für die Verwaltung der Strukturen.

Die zukünftige mögliche Erweiterungen des H3 Systems betrifft die Spanning Tree Erzeugung und den Darstellungsmodus.
Die Wahl eines Spanning Trees ist sehr wichtig für den visuellen Eindruck. Neue Algorithmen erlauben bessere Spanning Trees zu erzeugen, was nennenswert wäre.
Eine automatische Ausdehnung oder eine Kompression des Subgraphen gemäss Benutzerintersse ist in Vorbereitung.

Es bleibt noch ein grosses Problem bei riesen Graphen. Das H3 System auch mit einem mächtigen Navigationsystem, wird einfach nicht verständlich. Das heisst, wir müssen an ein anderes Konzept denken, besonders im Bereich der Graphenstrukturtheorie.

8 Literatur

1 Tamara Munzner, H3: Laying out large directed graphs in 3D hyperbolic space, Proceedings of the 1997 IEEE Symposium on Information Visualization, pages 2-10, 1997

2 Homepage von Tamara Munzner, http://graphics.stanford.EDU/~munzner/

3 Tamara Munzner and Paul Buchard, Visualization the structure of the World Wide Web in 3D hyperbolic space, Proceedings of the VRML '95 Symposium, pages 33-38, ACM SIGGRAPH, 1995

4 John Lamping, Ramana Rao and Peter Pirolli, A focus+context technique based on hyperbolic geometry for visualizing large hierarchies, Proceedings of CHI '95, ACM

5 Fsn 3D file manager system, http://www.sgi.com

6 Magnifind Filesystem manager, http://www.inxight.com

Ein spezieller Dank an Thomas Sprenger, Reto Lütolf und Remo Ziegler für ihre Hilfe…