



TECHNISCHE  
UNIVERSITÄT  
DRESDEN



INTERACTIVE  
MEDIA LAB  
DRESDEN

Fakultät Informatik Institut für Software- und Multimediatechnik, Multimedia-Technologie

Dokumentation

# INFOVIS

Paula Schöley, Robert Morawa, Gilbert Röhrbein, Alexandra Weiß

Eingereicht am 8.Juli 2013



# INHALTSVERZEICHNIS

1	Einführung . . . . .	4
1.1	Kanji Schriftsystem . . . . .	4
1.2	Tube Maps . . . . .	4
2	Motivation . . . . .	5
2.1	Warum Kanji Zeichen? . . . . .	5
2.2	Warum eine Tube Map für Kanji? . . . . .	5
3	Designprozess . . . . .	5
3.1	Auswahl eines passenden Datensatzes . . . . .	5
3.2	Entwicklung des Programmkonzepts . . . . .	6
3.3	Codierung visueller Variablen . . . . .	7
3.4	shneiderman mantra . . . . .	8
4	Herausforderungen . . . . .	8
4.1	Lösungsvorschläge . . . . .	8
5	Implementierung . . . . .	9
5.1	Entwicklungsumgebung . . . . .	9
5.2	Tubemap Layout . . . . .	9

# 1 EINFÜHRUNG

## 1.1 Kanji Schriftsystem

Die japanische Schrift besteht aus drei Alphabeten, wobei eines aus Logogrammen – sogenannten Kanji – besteht und die anderen beiden Silbenalphabete sind. Kanji stammen aus der chinesischen Sprache und wurden von den Japanern übernommen und haben sich seitdem parallel entwickelt. In einem Satz werden Kanji Zeichen unter Anderem für den Wortstamm und die Silbenalphabete für das „Auffüllen“ (beispielsweise Negation oder Zeitformen) verwendet. Hiragana und Katakana, also die Silbenalphabete, können aber auch als Ersatz für ein Kanji verwendet werden, indem sie die Aussprache (die Lesung) des Kanji beschreiben.

Kanji bestehen aus einzelnen Elementen, sogenannten Radikalen, die nach gewissen Regeln miteinander kombiniert werden können. Wenige der Radikale sind bereits bedeutungstragend, wie beispielsweise das Radikal 氵, welches für Kontext „Wasser“ steht. Somit ist es beispielsweise in den Wörtern für „Flut“ (氾) oder auch „Meer“ (海) enthalten.

## 1.2 Tube Maps

Tube Maps oder Transit Maps (deutsch: Liniennetzplan) sind eine Form der Visualisierung, die für Linien öffentlichen Nahverkehrs – also Busse, U- und Strass enbahnen – verwendet wird. Bisher werden Tube Maps nur von Menschen gefertigt, da viele verschiedene Faktoren das Aussehen und die Eigenschaften der Map beeinflussen.

### Eigenschaften

- Es existieren Stationen, die verschiedene Ausprägungen, wie „Endstation“ oder „Umstiegsmöglichkeiten“ besitzen.
- Stationen liegen auf Linien, die diese miteinander verbinden. Linien entsprechen den verlegten Gleisen beziehungsweise Routen, die die einzelnen Stationen verbinden. Es gibt die Möglichkeit, verschiedene Arten von Linien visuell zu unterscheiden, um zum Beispiel zwischen Straßenbahn- und Buslinie zu unterscheiden. Des Weiteren dürfen die Linien nur horizontal, vertikal oder im 45° Winkel zueinander sein.  
Linien können sich ab einer Station aufteilen, um zu zwei alternativen Endstationen zu fahren. Hinzu kommen Ringbahnen, bei denen die Endstation der Anfangsstation entspricht.
- In [Stott, 2008] werden als weiterer Bestandteil der Map topographische Metadaten, also Orientierungspunkte wie Flüsse, Flughäfen, Bahnhöfe, oder sehr wichtige touristische Orte genannt. Dabei helfen sie bei der Orientierung, indem geographische Bezüge zwischen Stationen und Landmarks hergestellt werden.
- Die Beschriftung der vorhandenen Stationen und Linien eines Liniennetzes spielt eine wichtige Rolle beim Kennenlernen und Verinnerlichen der vorhandenen Linien. Häufig wird die Farbcodierung der Linien für die Beschriftung der Linien verwendet, während dies bei Stationen nicht der Fall ist.
- Zonen stellen in Tube Maps meist Tarifzonen dar, können aber auch weitere Eigenschaften visualisieren, die sich durch Entfernung kodieren lassen.

### Weitere Verwendungszwecke

Wie in [Stott, 2008] beschrieben, gibt es auch weitere Anwendungsgebiete für Tube Maps, wie Projektpläne, Karten von Verbindungen zwischen Objekten wie beispielsweise Programmiersprachen, Aufbau von Webseiten, sowie Metabolic Pathways genannt. Eigene Recherche des Teams hat ähnliche Ergebnisse ergeben, wobei nicht immer bei allen Visualisierungen klar war, warum diese Form der Visualisierung gewählt wurde.

## 2 MOTIVATION

### 2.1 Warum Kanji Zeichen?

Die japanische Schrift ist streng hierarchisch und modular aufgebaut. Der Datensatz ist begrenzt, da keine neuen Kanji „erfunden“ werden, und er lässt sich reduzieren auf sogenannte Jouyou Kanji. Jouyou sind Zeichen, die besonders häufig im Alltagsleben verwendet werden, und die somit ein Großteil der Japaner beherrscht. Im Vergleich zur Gesamtzahl aller Kanji beschränken sich die Jouyou auf knapp 2000 Zeichen, was eine starke Reduzierung des Datensatzes ermöglicht. Diese Reduzierung ist nicht nur von praktischer Bedeutung, es ist außerdem für Interessierte, die die Sprache lernen wollen, ein guter Startpunkt.

### 2.2 Warum eine Tube Map für Kanji?

Durch den modularen Aufbau von Kanji können Radikale als Endstationen interpretiert werden, deren Linien durch mehrere Stationen (also Kanji) verlaufen. Dadurch ergibt sich der Aufbau eines Kanji anhand der Linien, die die Kanji-Station bedienen.

Lernende können somit existierendes Wissen über Radikale und Kanji zur Orientierung verwenden, oder aber neue Zeichen kennenlernen, indem sie der Linie eines bekannten Radikals folgen. Anhand der Häufigkeitsstatistik, die für die Alltags-Kanji vorhanden ist, kann ein Benutzer sich anhand häufiger Kanji darin enthaltenen Radikale merken und diese verwenden, um weitere Kanji zu lernen.

**Wer soll es am Ende verwenden?** Die Anwendung richtet sich primär an solche, die Kanji lernen möchten. Dabei wird davon ausgegangen, dass entweder ein spezielles Kanji gelernt oder nachgeschlagen wird, oder auch ein generelles Interesse an den Zeichen besteht. Soll das Kanji für „Baum“ gelernt werden, kann man weitere Kanji lernen, die sehr ähnlich dazu sind. [Rose, 2010] beschreibt eine Lernstrategie, die für Kanji eine besonders gute Anwendung findet. Bei dieser Strategie spielt neben visueller Ähnlichkeit auch das sogenannte „grouping“ (also die Bildung von Verknüpfungen zwischen Vokabeln basierend auf Bedeutung oder Konzepten) eine Rolle. Diese Erkenntnisse lassen vermuten, dass Tube Maps sich zum Lernen von Kanji eignen.

## 3 DESIGNPROZESS

### 3.1 Auswahl eines passenden Datensatzes

In der ersten Phase des Designprozesses hat das Team sich mit mehreren Datensätzen beschäftigt, die mit Hilfe einer Tube Map visualisiert werden können. Dabei wurden die Datensätze mittels Abstimmung auf vier reduziert, woraufhin sich jedes Teammitglied mit einem Thema beschäftigt hat. Die Ergebnisse wurden im Team vorgestellt und nach Abstimmung und Diskussion das Thema für die Visualisierung gewählt.

**Webseiten** Wie im Abschnitt 1.2 beschrieben lässt sich die Struktur einer Webseite mittels einer Tube Map darstellen. Diese Darstellung ist vor Allem für große und komplexe Webseiten von Vorteil, da ein Benutzer sich so einfach eine Übersicht über alle vorhandenen Inhalte verschaffen kann. Problematisch dabei ist jedoch, die Eigenschaften der Tube Map deutlich hervorzuheben und nicht einen Graphen als Ergebnis zu erhalten. Durch Verlinkungen zwischen verschiedenen Bereichen von Webseiten wird somit schnell eine der wichtigsten Eigenschaften von Tube Maps verletzt, nämlich die Zyklenfreiheit.

Die Idee des Teammitgliedes war daher, die häufig besuchten Webseiten und nicht die Struktur einer Webseite darzustellen. Oft besucht ein Benutzer einen Fundus von Seiten im Laufe des Tages. Diese Routine könnte der Linienführung der Linien entsprechen. Des Weiteren könnten Zonen für die grobe Klassifizierung des Inhalts der Seiten eingeführt werden, wie beispielsweise „Nachrichten“ oder „Zeitvertreib“. Dabei könnten Linien auch bestimmten Schlagwörtern, zum Beispiel „Technik“, zugeordnet werden, um zusammen mit den Zonen Seiten zu beschreiben, die primär über Neuigkeiten im Bereich von Technik und IT berichten.

**Bildeigenschaften** Ein weiterer Gedanke verfolgte das Ziel, die Eigenschaften von Bildern wie zum Beispiel Gemälden darzustellen. Hierbei würden die Linien Eigenschaften des Gemäldes und die Stationen das Gemälde selbst darstellen. Es wäre eine Möglichkeit, Muster in dem Gebrauch z.B. von Farbe unter Malern oder Fotografen im Zusammenhang mit ihrer Strömung, der Zeit oder des Bildthemas zu finden. Dabei würde die Tubemap sowohl auf X- und Y-Achse in Zonen aufgeteilt. Die X-Achse wäre in diesem Beispielkonzept eine Zeitachse, die Y-Achse weist die im Werk dominierenden Farben aus; sie würden den typischen Zonen einer Tubemap entsprechen.

Bei diesem Layout würden die Achsen je nach Inhalt gestreckt oder gestaucht und die Linien stetig von einer Hauptlinie abzweigen. Eigenschaften von Gemälden wie die Farbzusammensetzung könnten in Tortendiagrammen dargestellt werden, mehrere Gemälde mit gleichen Eigenschaften zu einer geclustert. Die Auswahl an Bildeigenschaften, die man so kodieren könnte, ist gross. Als Interaktionsmöglichkeiten böte sich an, Künstler oder Skalen zu wechseln sowie diverse Filter oder Neuskalierung vorzunehmen.

**Parallele Handlungsstränge** Ein dritter Ansatz beschäftigte sich mit parallel verlaufenden Handlungssträngen wie sie beispielsweise in der Fernsehserie Game of Thrones und dessen Buchvorlage vorkommen. Es würde Zuschauern oder Fans der Serie eine Möglichkeit geben, die bisherige Handlung zu rekapitulieren und bisherige Aufeinandertreffen von Charakteren darzustellen. Einzelne Hauptcharaktere erhalten hierbei eine Tubemaplinie, Kapitel bilden Stationen und Handlungsorte die Zonen der Tubemap. Diese würde entlang der X-Achse nach von links nach rechts verlaufen, grob an einem Zeitstrahl orientiert oder aber sich radial in alle Richtungen ausbreiten, wobei das die Vergleichbarkeit erschweren würde.

Stationlabels könnten in dieser Tubemap eine Wortgruppe als Zusammenfassung des Kapitels liefern. Man könnte die Stationen bei Point of View-Kapiteln den PoV auch in der Farbe des jeweiligen Charakters festhalten. Icons könnten Ereignisklassen wie Tode oder kämpferische Auseinandersetzungen symbolisieren. Interaktionen könnten hier die Filterung, das Eingrenzen eines Zeitbereiches und ähnliches beinhalten.

## 3.2 Entwicklung des Programmkonzepts

### Der gesamte Kanjiraum

Der erste Gedanke war, dass man die komplette Zahl Kanji anzeigen könnte. Aufgrund der hohen Zahlen an Radikalen würde jedoch ein wichtiges Element der Tubemaps entfallen; die Farbe der einzelnen Linien. Würde man jede Radikallinie einfärben, würden sich die Farben kaum voneinander abheben. Eine Idee war, das mittels Hervorhebung zu lösen, sodass man die Radikallinie selbst oder alle Linien eines Kanji per Selektion einfärben kann. Kleine Labels, die bei traditionellen Tubemaps genutzt werden, die Verkehrslinien innerhalb der Karte zu markieren, werden analog dazu genutzt, um die Radikallinien zu bezeichnen. Zur Unterscheidung von Kanji und Radikalen wird die Form der Station verwendet.

Innerhalb der Ansicht navigierte man vor allem mit Zooming und Panning. Eine Minimap, die die Position des Fensters auf dem Canvas angibt, sorgt hierbei für eine grobe Übersicht. Ideen wurden entworfen, die Kanji hier nach den in ihnen enthaltenen Radikalen zu clustern und diese Cluster erst mittels semantischem Zoom aufzulösen, um die überbordende Komplexität zu verringern und die Analyse nach dem Radikalkriterium zu erleichtern. Mit Filtern hat der Nutzer die Möglichkeit, die Menge der Kanji einzuschränken, hierfür eignen sich Attribute wie die unterschiedlichen japanischen Lesungen, die Bedeutungen, Strichzahl oder Häufigkeit des Kanji in japanischen Zeitungsartikeln. Eine Suche sollte Kanji anzeigen, die den ausgewählten Kriterien entsprechen, von dieser Suchliste wäre eine Autonavigation zum gesuchten Kanji möglich gewesen. Das Kanji wird für eine kurze Zeit farblich hervorgehoben, während alle Kanji, auf die die Suchkriterien zutreffen, mit dickerem Rand gezeichnet werden.

Mittels Hover hat der Nutzer die Möglichkeit, sich ein gewähltes Detail eines Kanji in Form eines Tooltips oder Stationslabels darzustellen. Da ein einfacher Hover bei einer komplexeren Tubemap rasch zu ungewünschten Details führen kann, wurde es mit einer kurzen Verweilzeit von ca. einer halben Sekunde kombiniert, sodass eine klare Eingabe des Nutzers vorliegen muss. Um weitere Vergleichbarkeit der Kanji herzustellen führten wir eine ausblendbare Detailtabelle ein. Dort wird pro Spalte ein ausgewähltes Kanji angezeigt, in den Zeilen die dazugehörigen Details. Auch hier sollte man automatisch zu den angezeigten Kanji navigiert werden.

## Die Central Station-Ansicht

Nach längerem Überlegen merkten wir, dass die bisherige Struktur des Konzeptes nicht zielführend ist oder sein wird. Man erhält nicht einmal einen groben Überblick über die Kanji, da die Struktur der Tubemap zu gross und komplex ist und man beim Hereinzoomen keinerlei Überblick über die restlichen Kanji erhält. Zudem fehlen unter anderem die für eine Tubemap charakteristischen farbigen Linien. Hinzu kam eine Überlegung, wie Kanji selbst gelernt werden. Normalerweise wird ein Kanji zum Lernen vorgegeben; mittels der Tubemap könnte man „verwandte“ Kanji, die dieselben Radikale teilen wie das zum Lernen ausgewählte Kanji. Auf diese Weise soll man das Kanji besser in seinen Kontext einbetten können.

In diesem Konzept gibt es einen „Hauptbahnhof“ (Central Station). Dieses Kanji wird in einer vorherigen Übersicht ausgewählt, in der alle Kanji gelistet sind und gefiltert werden können. Diese Central Station unterscheidet sich signifikant; sie ist grösser und zeigt alle Details des Kanji in der Station an. Es werden nur Linien zu den Radikalen angezeigt, die in dem Kanji enthalten sind. Da ein Kanji maximal neun Radikale enthält, ist es nun möglich die Linien mit Farben zu unterscheiden. Auch wird ein Gross teil der möglichen Überschneidungen auf diese Weise entfernt. Die Radikallinien führen durch alle Kanji, in denen sie enthalten sind.

Ansonsten bleiben viele Ideen des ursprünglichen Konzeptes erhalten. Farbige, kleine Radikale kennzeichnen die zugehörigen Linien. Kanji können in eine Detailtabelle eingetragen, durchsucht und gefiltert werden, ebenso die Autonavigation. Jedoch entfällt auf diese Weise das Hervorheben der Linien. Stationenbezeichner erfahren eine kleine Änderung; in der Central Station existiert nun die Möglichkeit, Detailkategorien auszuwählen, die dort angezeigt werden.

Zudem wird eine weitere Interaktionsmöglichkeit mit den Kanji eingeführt. Man kann nun ein Kanji anwählen, um es zu einer neuen Central Station werden zu lassen. Dabei sollen die Radikallinien, die bereits layoutet werden, weil sie in beiden Kanji enthalten sind, nur minimal verändert werden, während die neu hinzugekommenen Radikallinien ins Layout eingefügt werden. Nicht mehr benötigte Radikallinien werden ausgeblendet. Eine Liste der bisher ausgewählten Central Station Kanji wird in der Central Station selbst angezeigt.

## 3.3 Codierung visueller Variablen

Gemäss der in der Vorlesung „Interaktive Informationsvisualisierung“ besprochenen Effektivität visueller Variablen, werden in Bushu Mappu verwendet.

Visuelle Variable	Einsatz in Bushumappu
Linienlänge	keine, nicht charakteristisch für Tubemaps
2D-Position	keine
Ausrichtung	keine
Linienbreite	Rahmen gesuchter Kanji: dicker; breite Linienbündel: mehr gemeinsame Radikale
Grösse	Central Station: gross ; Station: normal
Form	Quadrat: Kanji; Kreis: Radikal
Krümmung	keine
Farbton	Radikal
Intensität	keine

### 3.4 shneiderman mantra

Overview

Zoom

Filter

Details-on-demand

Relate

History

Extract

## 4 HERAUSFORDERUNGEN

Da Haltestellen einen festen Ort haben, der sich nicht ändert, wird dieser als sehr wichtiger Faktor („initial embedding“) für das Layouting in Automatic Layout of Metro Maps using Multicriteria Optimisation von Stott ([Stott, 2008]) beschrieben. Dabei kann sich die Position geringfügig verändern, zum Beispiel werden Entfernungen zwischen Haltestellen auf dem Plan teilweise nicht in dem eigentlich vorhandenen Abstand dargestellt, sondern der Abstand wird angeglichen, um ein uniformes Aussehen des Plans zu gewährleisten. Diese Verortung existiert bei Kanji in diesem Masse nicht und daher müssen Richtungen und Positionen durch einen Algorithmus festgelegt werden. Ähnlich verhält es sich mit der Kreuzung von Linien außerhalb von Haltestellen. Vor allem bei Schienenverkehr findet eine solche Kreuzung selten statt, meist bedienen Linien stattdessen mindestens eine gemeinsame Haltestelle. Da eine physische Begrenzung dieser Art bei dem verwendeten Datensatz nicht vorhanden ist, müssen Linien dort aufwendig per Routingalgorithmus gelegt werden, um Überschneidungen soweit möglich zu vermeiden.

### 4.1 Lösungsvorschläge

Da der Datensatz mit knapp 2000 Kanji deutlich grösser ist, als die Haltestellen der Dresdner Verkehrsbetriebe([DVB-AG, 2011])

**Clustering** Bei der Linearen Anordnung werden alle Radikale auf einer x-Achse (wie beim kartesischen Koordinatensystem) angeordnet und über ihnen alle Kanji. Diese können nach Kriterien wie Strichanzahl, Schuljahr oder Häufigkeit auf der y-Achse sortiert werden. Weiterhin sollten diese grob in Cluster aufgeteilt werden, um sie nahe an den Radikalen zu halten, aus denen sie bestehen beziehungsweise zu denen sie gehören. Dies bedeutet, dass die Kanji je nach Menge gleicher gemeinsamer Radikale einem Cluster zugeordnet werden.

Hierbei sollte zwischen jedem Kanji ausreichend Platz gelassen werden, um im Zweifelsfall alle 237 Linien zwischen zwei Kanji hindurch zu führen. Ist das Routing zwischen allen Kanji beendet, so können einige dieser Lücken wieder verkleinert werden, vorausgesetzt, dass sich dadurch nicht (noch mehr) Linien kreuzen. Sehr nachteilig dabei ist der grosse Platzbedarf, der sich durch grosse ungenutzte Zwischenräume ergibt. Des Weiteren hat sich gezeigt, dass Clusterbildung nicht ausreicht um Überschneidungen von Linien zu verhindern.

Um diese Probleme zu umgehen hat das Team mit einer radialen Anordnung von Radikalen experimentiert, um freie Fläche besser zu nutzen, aber dieser Ansatz brachte keine besseren Ergebnisse hervor und wurde deshalb nicht weiter verfolgt.



**Reduktion des Datensatzes** Eine drastische Reduktion des gerade angezeigten Datensatzes wurde dadurch erreicht, dass nur die Radikale eines ausgewählten Kanji gezeigt werden. Auf den Radikal-Linien befinden sich dann dem ausgewählten Kanji alle Kanji, in denen das Radikal vorkommt. Durch diese Reduktion lassen sich die (maximal neun) Linien strahlenförmig anordnen, wobei Verbindungen zwischen Linien existieren können. Diese werden mit einer Ecke versehen, um möglichst wenige Überschneidungen von Linien zu erhalten.

## 5 IMPLEMENTIERUNG

Die Software ist eine vollständig Client-seitige Webanwendung. Diese Entscheidung wurde in Abwägung gegenüber C++ mit Qt 5 getroffen. Ausschlaggebende Argumente waren die Verfügbarkeit von CSS und SVG zur Darstellung im Gegensatz zu imperativen und Objekt-orientierten Ansätzen sowie das einfache Zeigen und Teilen mit anderen Menschen über einen Link und einen Browser.

### 5.1 Entwicklungsumgebung

Mit der Entscheidung zur Webanwendung war die erste Frage, welche Programmiersprache die vorherrschende sein sollte. Zwischen JavaScript, CoffeeScript, Java und Haxe fiel die Wahl auf CoffeeScript 1.6.2. Stark typisierte Sprachen, die zu JavaScript compilieren, wie Java und Haxe, haben das Problem, dass JavaScript Bibliotheken wie d3.js, einen abschreckenden Aufwand erfordern, die Interfaces zu definieren, so dass der Vorteil der starken Typisierung auch genutzt werden kann. CoffeeScript ist JavaScript in sofern überlegen, da es ein Klassen-Pattern bereits in die Sprache integriert und auch vor anderen Unklarheiten in JavaScript schützt.

Zur Darstellung selbst wird SVG 1.1 und CSS 2010 verwendet, um mit Hilfe von CSS schnell schicke Effekte und Transitionen nach Bedarf zu definieren und sie vom Browser bereits effizient umgesetzt werden. CSS Effekte sind jedoch nur auf DOM Elemente anwendbar, was wiederum SVG verlangt als Darstellungsframework. d3.js 3.1.6 ist darauf ausgelegt DOM Elemente basierend auf Daten zu erstellen. Andere Darstellungsbibliotheken für Webanwendungen legen entweder ihren Fokus nicht auf SVG und CSS oder sind nicht Daten-gebunden.

Zwischen Firefox 23 und Chromium 28 fiel die Wahl auf letzteren, da der Chromium SVG mit Text effizient darstellt. Der Firefox stößt hier an eine Grenze.

Weitere verwendete Technologien sind require.js 2.1.6 um die Software in Module zu teilen, reactor.js commit cdbf994 als marginale Abhängigkeit, die lediglich für einen simplen Fall mal ausprobiert wurde, GNU Make 3.81 um typische Entwicklungsprozesse zu automatisieren und Python 3.2.3 als lokaler Webserver.

### 5.2 Tubemap Layout

Die Software für ein ansehnliches Layout für die Tubemap zu implementieren, gestaltete sich als sehr schwierig. Die Doktorarbeit von [Stott, 2008] nahmen wir als Inspiration, jedoch eignete sich diese Arbeit lediglich für einen groben Ansatz. Im Detail mussten viele andere Entscheidungen getroffen werden und andere Ansätze ausprobiert werden.

Prinzipiell besteht der Layout Prozess aus zwei Phasen. Zuerst wird eine initiale Einbettung bestimmt. Dabei sollen Positionen der Knoten bestimmt werden und wie diese untereinander verbunden sind. Danach wird der entstandene Graph optimiert. Dies geschieht durch eine kontinuierliche Bewertung und Veränderung um eine bessere Bewertung zu erzielen. Die Bewertung richtet sich nach mehreren Kriterien und Regeln.

Konkret ist das Ergebnis der initialen Einbettung, dass das zentrale Kanji in der Mitte ist. Die Radikale des zentralen Kanji werden hier als relevante Radikale bezeichnet. Von dem zentralen Kanji ausgehend, denkt man sich für jedes relevante Kanji Strahlen in 90° und 45° Winkel zueinander. Für jedes relevante Radikal werden die Kanji genommen, die dieses Radikal enthalten und eingeteilt in

solche, die von allen relevanten Radikalen nur eins enthalten (Lo-Kanjis), und solche, die mehrere relevante Radikale enthalten (Hi-Kanjis). Hi-Kanjis werden unter den Strahlen verteilt, Lo-Kanjis am entsprechenden Strahl hinten angefügt. Alle Kanjis haben nun eine Position. Um Kanjis mit einander zu verbinden, werden Verbindungen nacheinander gezogen für jedes relevante Radikal. Vom zentralen Kanji aus werden alle Hi-Kanjis des zum Radikal zugehörigen Strahls verbunden und anschliessend wird jeder Strahl durchgegangen und alle dortigen Hi-Kanjis, die das aktuelle Radikal enthalten, nacheinander verbunden. Zum Schluss wird noch eine Verbindung zu den dazugehörigen Lo-Kanjis gezogen.

Die initiale Einbettung ist ausgesprochen wichtig für eine erfolgreiche Optimierung. Die Optimierung nach mehreren Kriterien wird von [Stott et al., 2011] beschrieben. Ein Teil dieser Optimierung ist in der vorliegenden Software implementiert, jedoch in entscheidenden Aspekten abgeändert um schneller zu einem guten Ergebnis zu kommen, die Entwicklung zu vereinfachen und auf spezielle Bedürfnisse der Kanji-Tubemap einzugehen. Die Optimierung findet parallel in einem Worker statt um die Interaktivität der Anwendung nicht zu behindern. Kontinuierlich wird ein Knoten im Graph ausgewählt und geprüft ob in der Umgebung eine Position für den Knoten ist, bei der die Bewertung des Graphen besser wird. In die Bewertung fliessen Regeln und Qualitätskriterien ein. Weniger Regeln zu verletzen ist stets besser als eine höhere Qualität zu erreichen.

Folgende Regeln möglichst nicht verletzt werden:

- keine Kanten, die unter einem Knoten durchgehen, die nicht zu dem Knoten selbst gehören (**wrongEdgesUnderneath**)
- der Knoten darf dem Knoten für das zentrale Kanji nicht zu nahe kommen (**tooNearCentralNode**)
- keine Kanten, die andere Kanten kreuzen (**edgeCrossings**)

Folgende Qualitätskriterien flossen in die Version der Software ein, die zur finalen Präsentation gezeigt wurde:

- aneinander liegende Kanten sollen möglichst gerade sein (**lineStraightness**)
- Kanten sollen möglichst kurz sein (**lengthOfEdges**)

# LITERATURVERZEICHNIS

[DVB-AG, 2011] DVB-AG (2011). Website.

[Rose, 2010] Rose, H. (2010). Kanji learning of Japanese language learners on a year-long study exchange program at a Japanese university: An investigation of strategy use, motivation control and self-regulation. PhD thesis, University of Sydney.

[Stott et al., 2011] Stott, J., Rodgers, P., Martinez-Ovando, J. C., and Walker, S. G. (2011). Automatic metro map layout using multicriteria optimization. Visualization and Computer Graphics, IEEE Transactions on, 17(1):101–114.

[Stott, 2008] Stott, J. M. (2008). Automatic Layout of Metro Maps using Multicriteria Optimisation. PhD thesis, University of Kent.