

Informationsvisualisierung

Übungspräsentation

Jennifer Kane, Jonas Gröger

December 1, 2014

3.1 Visuelle Eigenschaften von Graphen

Layout

- ▶ **Arten:** hierarchische, kreisförmige, kraftbasiert
- ▶ **Beispiel:** GPS-Rohdaten → POIs → Werbung
- ▶ **Begründung:** Isomorphie von kraftbasierten Graphen wird ausgenutzt: Orte nicht Laufwege für Werbung interessant

3.1 Visuelle Eigenschaften von Graphen

Visuelle Codierung

- ▶ **Arten:** Form, Größe und Farbe eines Knoten / einer Kante
- ▶ **Beispiel:** Viele Freunde → großer Knoten
- ▶ **Begründung:** Visuell schnell aufzufassen

3.1 Visuelle Eigenschaften von Graphen

Rendering

- ▶ **Arten:** Interaktion im Allgemeinen
- ▶ **Beispiel:** Warmwasserleitungen zwischen Städten (Übersicht) und innerhalb einzelner Städte (Detailansicht) durch zoomen erfahrbar
- ▶ **Begründung:** Ansicht auf geeignetem Abstraktionsniveau

3.1 Visuelle Eigenschaften von Graphen

Vereinfachung

- ▶ **Arten:** Cluster, collapsing, Hierarchien
- ▶ **Beispiel:** Dateisysteme, z.B. Finder, Nautilus, Explorer
- ▶ **Begründung:** Ansicht auf geeignetem Abstraktionsniveau

3.2 Visual Clutter

Kamada-Kawai - Rahmenbedingungen

- ▶ ungerichtete, gewichtete Graphen
- ▶ gerade Kanten, keine Beschränkung bei Positionierung der Knoten
- ▶ zusammenhängender Graph (sonst: Zusammenhangskomponenten einzeln zeichnen)

3.2 Visual Clutter

Kamada-Kawai - Kriterien

- ▶ Anzahl Kantenkreuzungen reduzieren
- ▶ gleichmäßige Verteilung von Knoten und Kanten (wichtiger für menschliches Verständnis)

3.2 Visual Clutter

Kamada-Kawai - Idee

- ▶ Stahlringe werden durch Federn zusammengehalten
- ▶ Ziel: Minimierung der Energie dieses Systems
- ▶ ideale Distanz zwischen zwei Knoten proportional zur Länge eines kürzesten Pfades zwischen ihnen

3.2 Visual Clutter

Kamada-Kawai - Algorithmus

- ▶ Berechnung der Distanz $d_{i,j}$ (Shortest Path Algorithm - Floyd), in gewichteten Graphen: Distanz = Summe der Gewichte
- ▶ Berechnung der Länge $l_{i,j}$ (aus Distanz und optimaler Länge)
- ▶ Berechnung der Federstärke $k_{i,j}$ (aus Distanz)
- ▶ Bestimmung der initialen Knotenpositionen
- ▶ Schrittweise Minimierung der Energie: Verschieben eines Knotens in stabile Position

Laufzeit: $O(|V|^3 + T * |V|)$, T = Anzahl der Iterationen

3.2 Visual Clutter

Kamada-Kawai - Ergebnisse

- ▶ symmetrische Graphen werden auch symmetrisch angeordnet
- ▶ isomorphe Graphen werden auf dieselbe Art und Weise dargestellt (evtl. verschoben, gedreht oder gespiegelt)

3.2 Visual Clutter

Fruchtermann-Reingold - Rahmenbedingungen

- ▶ ungerichtete Graphen
- ▶ gerade Kanten, Graph begrenzt durch festen Rahmen

3.2 Visual Clutter

Fruchtermann-Reingold - Kriterien

- ▶ gleichmäßige Verteilung von Knoten und Kanten innerhalb des Rahmens
- ▶ Anzahl Kantenkreuzungen reduzieren
- ▶ Streben nach symmetrischer Darstellung
- ▶ Nachbarknoten sollen nah beieinander liegen, Knoten sollen aber nicht zu eng zusammen gezeichnet werden

3.2 Visual Clutter

Fruchtermann-Reingold - Idee

- ▶ Knoten verhalten sich wie Atomteilchen und üben anziehende und abstoßende Kräfte aufeinander aus
- ▶ zur Vereinfachung: anziehende Kräfte nur zwischen Nachbarknoten, abstoßende Kräfte zwischen allen Knoten
- ▶ Geschwindigkeit statt Beschleunigung, statisches Gleichgewicht statt dynamischem Gleichgewicht

3.2 Visual Clutter

Fruchtermann-Reingold - Algorithmus

- ▶ Berechnung der abstoßenden Kräfte
- ▶ Berechnung der anziehenden Kräfte
- ▶ Verschiebung begrenzen durch Temperatur
- ▶ Teilchen am Rahmen stoppen
- ▶ Sonderfall: zwei Knoten mit derselben Position → behandeln als wären sie dicht beieinander, dann wirken große abstoßende Kräfte

Laufzeit: $O(T * (|V|^2 + |E| + |V||E|))$,

$T = \text{Anzahl der Iterationen} = O(|V|) \text{ oder } O(|E|)$

3.2 Visual Clutter

Fruchtermann-Reingold - Algorithmus (Variante)

- ▶ Einteilung in Gitter
- ▶ abstoßende Kräfte nur zwischen Knoten benachbarter Quadrate
- ▶ besser: abstoßende Kräfte zwischen Knoten innerhalb eines Kreises um Ausgangsknoten

Laufzeit: $O(T * (|V| + |E| + |V||E|))$,

$T = \text{Anzahl der Iterationen} = O(|V|) \text{ oder } O(|E|)$

3.2 Visual Clutter

Fruchtermann-Reingold - Ergebnisse

- ▶ symmetrische Graphen werden besonders gut dargestellt
- ▶ isomorphe Graphen werden auf dieselbe Art und Weise dargestellt (evtl. verschoben, gedreht oder gespiegelt)
- ▶ relativ viele Kantenkreuzungen bei Bäumen, Wurzel nicht zu erkennen
- ▶ sehr unterschiedlich gute Ergebnisse bei dreidimensional wirkenden Graphen
- ▶ Variante: Dreidimensionales Layout
 - ▶ Layouting in drei Dimensionen, Projektion in die Ebene
 - ▶ keine Verbesserung der Ergebnisse im Vergleich zum 2D-Algorithmus
 - ▶ Auswahl der Projektion durch Benutzer
- ▶ Zusammenhangskomponenten driften auseinander und verbreiten sich am Rand → einzelne Berechnungen (wie bei Kamada-Kawai) oder Gitter-Variante
- ▶ sehr schnelle Berechnungen

3.2 Visual Clutter

Gemeinsamkeiten & Unterschiede

- ▶ Layouting ungerichteter Graphen mit geraden Kanten, Kamada-Kawai unterstützt auch gewichtete Graphen, Fruchtermann-Reingold ordnet Graphen innerhalb eines festen Rahmens an
- ▶ Ziel beider Verfahren: gleichmäßige Verteilung der Knoten und Kanten, Kamada-Kawai dabei eher fixiert auf Vermeidung von Kantenkreuzungen
- ▶ verschiedene Ideen liegen zugrunde: System aus Federn \leftrightarrow Kräfte zwischen Atomteilchen
- ▶ verschiedene Laufzeiten, Fruchtermann-Reingold wesentlich schneller
- ▶ gute Darstellungen symmetrischer und isomorpher Graphen, Fruchtermann-Reingold Probleme mit komplexeren Graphen

3.3 Graphen übersichtlicher gestalten

Gruppieren von Knoten und Kanten

- ▶ mehrere Knoten zu einem Knoten gruppieren
- ▶ entsprechende Kantenverbindungen
- ▶ Beispiel: Netzwerk verteilt über die gesamte Welt → Einheiten nach Ländern gruppieren → Beziehungen zwischen den Ländern darstellen

3.3 Graphen übersichtlicher gestalten

Einfärben von Knoten oder Kanten

- ▶ thematische Gruppierungen durch verschiedene Farben kennzeichnen
- ▶ schneller Überblick
- ▶ Erkennen interner und externer Beziehung dieser Gruppen

3.3 Graphen übersichtlicher gestalten

Verschiedene Knotendarstellungen

- ▶ Knoten durch verschiedene Symbole darstellen
- ▶ allgemeine Formen: Quadrate, Kreise, Dreiecke, Sterne, ...
- ▶ Darstellungen, die direkt Assoziationen zur jeweiligen Gruppe hervorrufen (z.B. Gendersymbole)

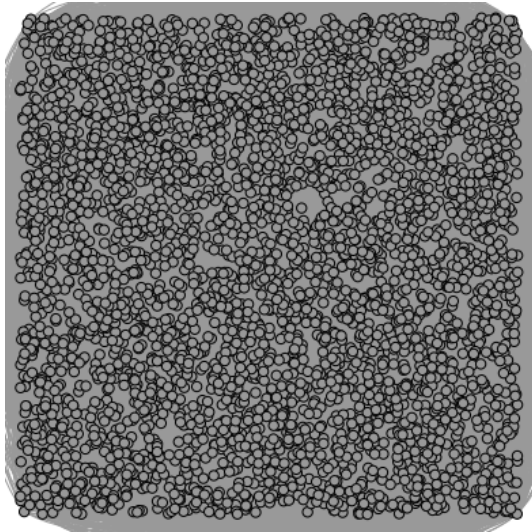
3.3 Graphen übersichtlicher gestalten

Interaktionstechniken

- ▶ Reduzieren kognitiver Belastung
- ▶ spielerische Analyse
- ▶ Beispiele: zoomen, verschieben, visuelle Hervorhebungen, ...

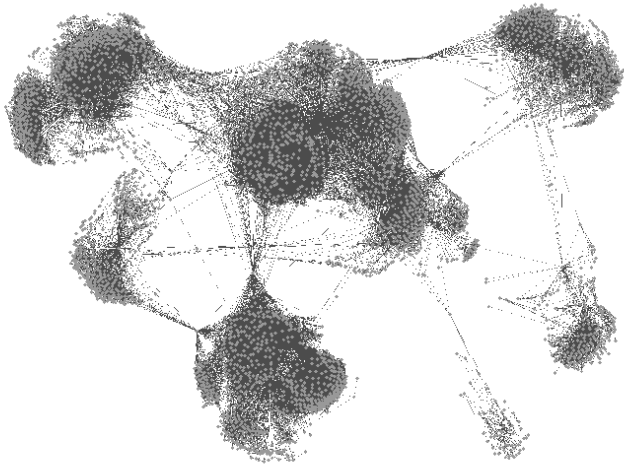
3.4 Visualisierung mit Gephi

Grafik direkt nach Import der Dateien



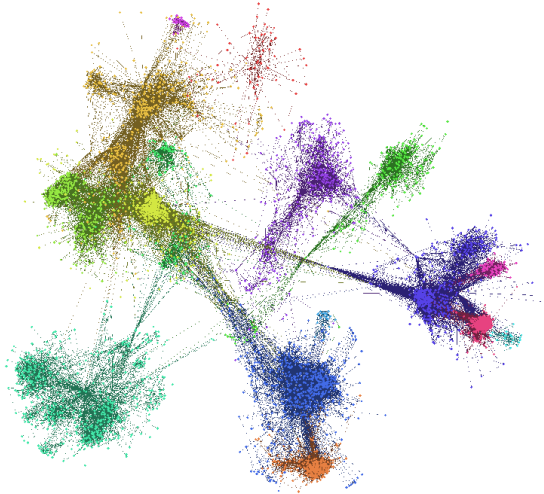
3.4 Visualisierung mit Gephi

Grafik nach Anwendung von Layout-Algorithmen



3.4 Visualisierung mit Gephi

Grafik nach Anwendung von Layout-Algorithmen und Einfärben der Knoten



Vielen Dank für die Aufmerksamkeit.