Informationsvisualisierung und Visual Analytics

Zusammenfassung

Fabian Damken

10. März 2022

TECHNISCHE UNIVERSITÄT DARMSTADT

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Liste der Algorithmen

1 Einleitung

In dieser Zusammenfassung werden zwei Themen behandelt: Informationsvisualisierung und Visual Analytics. Dabei sollen die Frage beantwortet werden, wie verschiedene Daten *gut* visualisiert werden können und wie Visualisierung die Analyse unterstützen können. Viele Ideen der folgenden Kapitel und grundlegender Techniken bauen dabei auf dem Verständnis grundlegender Prozesse des Gehirns ab. Denn: Eine Visualisierung soll dem Gehirn des*der Nutzer*in Arbeit abnehmen. Dabei sollen für jede Visualisierung die folgenden drei Fragen beantwortet werden:

- Was wird wie dargestellt?: Formale Beschreibung einer Visualisierung.
- Was ist gut und (möglichst) ohne Anstrengung sichtbar?: Prinzipien der Wahrnehmung kennen und anwenden.
- Was hilft dem*der Nutzer*in bei der Aufgabe?: Beschreibung und Wahrnehmungsprinzipien im Kontext einer Aufgabe bewerten.

Ein relevantes, bisher noch nicht erwähntes, Wort in den obigen Fragen ist die *Aufgabe*. Zu Beginn jeder Visualisierung muss zunächst die *Aufgabe* der Visualisierung identifiziert werden. Dies sind oftmals Entscheidungen, die (objektiv) durch Daten und Informationen getroffen werden (sollten). Dies ist in Abbildung 1.1 dargestellt. In der Praxis werden jedoch häufig einfach Visualisierungkataloge (große Datenbanken mit Visualisierungstechniken) nach einer „schönen“ Visualisierung durchsucht. Dadurch wird das Dasein der Visualisierung als Werkzeug jedoch zu dem Zweck gemacht, d. h. die Visualisierung wird ein Selbstzweck. Dies sollte eigentlich nie der Fall sein!

Oft die Aussage getroffen, dass „ein Bild mehr sagt als tausend Worte.“ Im Allgemeinen sollte allerdings eher gesagt werden, dass ein Bild etwas *anderes* als tausend Worte sagt: Sprachliche Artefakte (wozu auch Zahlen gehören), werden von dem Gehirn *bewusst* und verarbeitet und oftmals in eine zeitliche Reihenfolge gebracht. Eine Visualisierung der selben Daten hingegen ist ein bildliches Artefakt und erlaubt eine *unbewusste* Verarbeitung, bei der die Informationen im Raum strukturiert werden. Dadurch können komplexe Zusammenhänge sehr schnell vermittelt und erfasst werden.

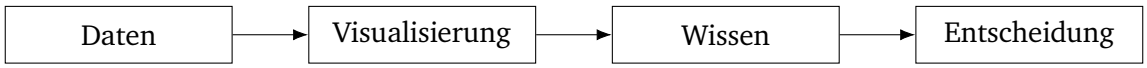

Abbildung 1.1: Eine Visualisierung dient immer der Erfüllung einer Aufgabe und soll zu einem Erkenntnisgewinn führen. Oftmals steht am Ende davon eine Entscheidung, die objektiv durch Daten getroffen werden soll. Bei der Erstellung einer Visualisierung sollte dieser Prozess also rückwärts durchgeführt werden, d. h. ausgehend von der Frage, welche Entscheidung getroffen werden soll.

1.1 Anwendungen von Visualisierungen

Die Anwendung einer Visualisierung, lässt sich in zwei Kategorien einteilen: *Erfassen* und *Produzieren* von Informationen, wobei erstere durch Informationsvisualisierung und letztere durch Visual Analytics „bearbeitet“ werden.

Innerhalb der Informationsvisualisierung werden die folgenden Gruppen unterschieden:

- *Explain:* Es sollen bekannte Informationen an andere vermittelt werden (möglicherweise, aber nicht immer, interaktiv).
- *Explore:* Es sollen neue Information auf Basis von Daten gefunden oder unsichere Informationen bestätigt werden (üblicherweise sehr interaktiv; das Ziel ist nicht immer bekannt).
- *Enjoy:* Zwanglose und durch Neugier getriebene Begegnung mit den Daten; dabei ist die Aufgabe selten bekannt.

Von diesen drei Arten der Informationsvisualisierung werden in dieser Zusammenfassung vor allem die ersten beiden behandelt.

Innerhalb der Visual Analytics werden die folgenden Gruppen unterschieden:

- *Annotate*
- *Record*
- *Derive*

1.2 Identifizierung der Visualisierungsaufgabe

Bei der Identifizierung der Aufgabe sollte auch mit einbezogen werden,

- welche Informationen als bekannt vorausgesetzt werden,
- welche Informationen gesucht werden, und
- was mit den neuen Informationen gemacht wird oder gemacht werden soll.

Das Design der Visualisierung bestimmt damit essentiell, wie gut mit der Visualisierung gearbeitet werden kann durch Wiedererkennung bekannter Informationen und Erkennung neuer Informationen. Die erste Regel ist dabei, wie bereits erwähnt, das die Visualisierung ein Werkzeug und kein Selbstzweck ist. Das lässt sich wie folgt zusammenfassen:

Have something to tell or something to ask!

1.3 Negativbeispiele

2 Der Informationsvisualisierungsprozess

Der *Informationsvisualisierungsprozess*, dargestellt in Abbildung 2.1, wurde von Card et al. als Referenzmodell entworfen, um die Erstellung einer Visualisierung zu strukturieren. Dabei wird zwischen den Repräsentationen (*Rohdaten*, *Datentabellen*, *Visuelle Strukturen* und *Views*) unterschieden. Der Übergang zwischen den Strukturen wird durch die *Datentransformation*, die *Visuelle Abbildung* und die *View Transformation* beschrieben. Eine weitere Zentrale Komponente ist der*die Nutzer*in (*Mensch*), welche*r die einzelnen Schritte modifiziert. Dadurch wird das Modell zu einem Kreislauf der Visualisierung. In diesem Kapitel wird jeder Schritt separat behandelt, beginnend mit der Datentransformation (Abschnitt 2.1) über die visuelle Abbildung (Abschnitt 2.2) bis zur Interaktion (Abschnitt 2.4). Im folgenden wird eine kurze Übersicht über die Transformationsschritte gegebene:

- *Datentransformation:* Der der *Datentransformation* werden die *Rohdaten* in *Datentabellen* umgewandelt. Diese können dann in den nächsten Schritten einfacher als die Rohdaten verarbeitet werden, da diese häufig ungeordnet und unstrukturiert vorliegen.
- *Visuelle Abbildung:* Bei der *visuellen Abbildung* findet der Hauptteil der Visualisierung statt. Dabei werden einzelne Datenvariablen auf visuelle Attribute (bspw. die Position) abgebildet, was die *visuellen Strukturen*.
- *Datentransformation:* Abschließend werden in der *View Transformationen* die tatsächlichen Datenwerte auf die Variablenwerte der visuellen Struktur abgebildet, woraus sich die tatsächliche Visualisierung, die *View*, ergibt.

Die Unterscheidung zwischen der visuellen Abbildung und der View Transformation ist, dass ersteres beschreibt, *dass* eine Datenvariable auf eine visuelle Variable abgebildet wird, und letzteres beschreibt, *welcher* konkrete Datenpunkt auf welchen konkreten Wert abgebildet wird.

Der letzte Schritt im Informationsvisualisierungsprozess ist die Interaktion, welche durch die zurückgeführten Pfeile in Abbildung 2.1 dargestellt wird. Dies ist die technische Möglichkeit, die einzelnen Schritte zu verändern und anzupassen. Es soll dem*der Nutzer*in dabei möglich sein, einzelne Komponenten anzupassen, ohne auf eine neue Visualisierung des*der Designer*in zu warten. Für den*die Designer*in kann dadurch zu teilen das Problem gelöst werden, dass eventuell die genaue Aufgabe noch nicht bekannt ist.

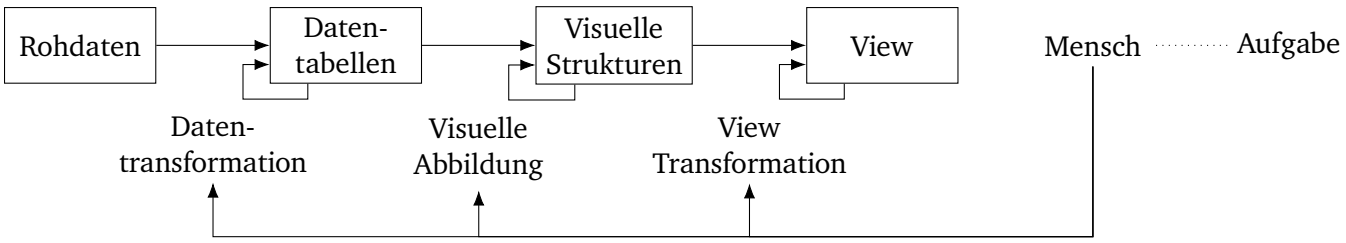

Abbildung 2.1: Der Informationsvisualisierungsprozess nach Card et al.

2.1 Die Datentransformation

Bei der Datentransformation werden die Rohdaten in Datentabellen umgewandelt, welche in den folgenden Schritten einfacher zu verarbeiten sind. Außerdem können eine Reihe an Vorverarbeitungstechniken (bspw. Datenbereinigung) angewandt werden. Dieser Abschnitt führt zunächst in die grundlegenden Datentypen und -strukturen sowie anschließend verschiedene Techniken der Vorverarbeitung ein.

2.1.1 Datentypen und Datenstrukturen

Zur Diskussion und Abbildung von Daten auf visuelle Strukturen ist es zunächst sinnvoll, die verschiedenen auftretenden Datentypen und -strukturen zu betrachten. Die Unterscheidung zwischen einem Typ und einer Struktur wird dabei grundlegend durch die Anzahl der zugeordneten Datenvariablen getroffen: Einfache Datentypen beziehen sich immer auf genau eine Datenvariable während Datenstrukturen Beziehungen zwischen mehreren Datenvariablen (welche meist Eigenschaften von Objekten, auch *Items* genannt, sind) und Datensätzen beschreiben. Eine Visualisierung kann anschließend eine beliebige Kombination aus Datenvariablen, Items und Beziehungen abbilden.

Datentypen

Grundlegend wird zwischen *nominalen*, *ordinalen* und *quantitativen* Datentypen unterschieden. Bei einem nominalen Datentyp ist dabei zwischen zwei Werten nur *Gleichheit* und *Ungleichheit* definiert. Dies können z. B. Personen, Länder, etc. sein. Sind nur nominelle Daten zu visualisieren, kann eine Tabelle hier tatsächlich die beste Wahl sein! Ordinale Datentypen haben zusätzlich zu (Un-) Gleichheit noch eine *Ordnung*, bspw. Schulnoten oder Rangordnungen. Der in Anzahl Operationen „mächtigste“ Datentyp sind quantitative Daten, die auch noch *Differenzen* und *Differenzverhältnisse* haben. Ferner wird bei quantitativen Daten zwischen *diskreten* und *kontinuierlichen* Daten. Außerdem gibt es *Intervall-* und *Verhältnisskalen*: Auf ersterer sind Verhältnisse nicht vernünftig berechenbar, da die Skala keinen natürlichen Nullpunkt hat (bspw. Temperatur in °C). Letztere hat einen natürlichen Nullpunkt, weshalb auch Verhältnisse auf Werten Sinn ergeben (bspw. Temperatur in Kelvin). Des weiteren gibt es *lineare* und *zyklische* Skalen, bei der die Ordnung eine Totalordnung, bzw. eine „künstliche“ Ordnung, ist.

Die unterschiedlichen Datentypen sind in Abbildung 2.2 zusammengefasst. Im Allgemeinen lässt sich also sagen, dass sich Datentypen vor allem durch die auf ihnen ausführbaren Operatoren unterscheiden. Außerdem ist nicht die Repräsentation, sondern die Bedeutung des Datentyps relevant (bspw. sehen Schulnoten aus wie Zahlen, sind aber ordinale Daten).

Datenstrukturen

Die bisher diskutierten Daten*typen* beziehen sich immer auf genau eine Datenvariable, wohingegen Daten*strukturen* auch Zusammenhänge zwischen Daten abbilden. Dabei sind Datenvariablen oft Eigenschaften von Objekten, die hier *Item* genannt werden.

Die einfachste Form einer Datenstruktur ist eine Tabelle, die außerdem sehr flexibel ist, weshalb sie im Informationsvisualisierungsprozess (Abbildung 2.1) namensgebend ist. Innerhalb einer Tabelle hat jede Spalte eine von zwei Rollen: Key oder Value. Dabei definieren die Datentype der Keys die eigentliche Datenstruktur und die Values sind von den Keys abhängig. Dabei müssen die Keys eine Zeile eindeutig identifizieren, d. h. es darf keine Kombination mehrfach auftreten.

Hat eine Tabelle keine Keys sondern ausschließlich Attribute, so werden die Items nur technisch Unterschieden (bspw. durch eine Zeilennummer) und die Daten sind „rein“ uni-/bi-/multivariat. Dies ist oft der Fall bei

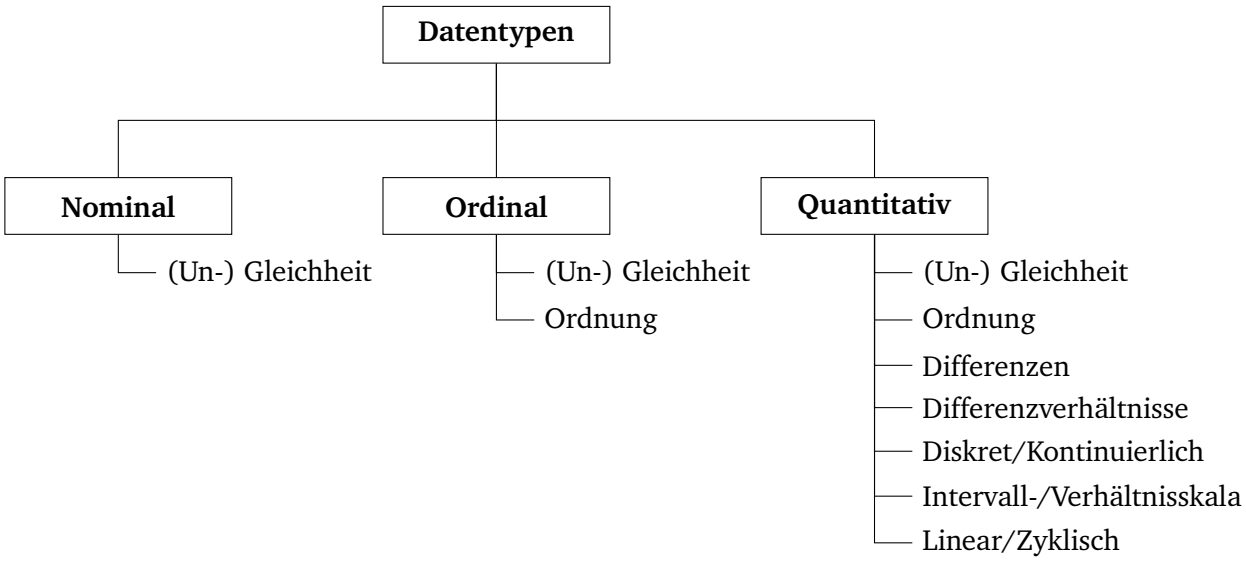

Abbildung 2.2: Unterschiedliche Datentypen

Daten, bei denen die Eigenschaften wichtiger sind als die Identifikation oder bei Daten, in denen die Items nicht identifiziert werden dürfen (Anonymisierung).

Da eine Tabelle sehr flexibel ist, sind die folgenden Datenstrukturen allesamt auf eine Tabelle abbildbar (die Abbildung ist jeweils durch die Angabe der Keys gegeben). Darüber hinausgehend gibt es auch Datenstrukturen, die am besten auf mehrere Tabellen (als relationales Modell) abgebildet werden. Achtung: Zwar ist die Darstellung als Tabelle hier zweckmäßig, jedoch müssen die Daten nicht zwangsweise technisch als Tabelle gespeichert werden!

Zeitbezogene Daten Zeitbezogene Daten haben mindestens einen Zeitstempel als Key, welcher sowohl relativ als auch absolut sein kann. Die Zeit wird häufig über die Position Achse visualisiert (z. B. Corona-Fallzahlen). Die Darstellung zeitbezogener Daten wird intensiv in Abschnitt 3.3 behandelt.

Ortsbezogene Daten Bei ortsbezogenen Daten ist mindestens ein Key eine Ortsangabe, die beispielsweise als geographische Position (Längen- und Breitengrad) oder Ortsname realisiert werden kann. Eine beliebte Darstellung ist das Anzeigen auf einer Karte.

Bewegungsdaten Bewegungsdaten stellen eine Kombination aus zeit- und ortsbezogenen Daten dar. Auch hierbei gibt es mehrere Varianten, bspw. können Messungen an mehreren, aber individuell festen Orten, durchgeführt werden (z. B. Wetterstationen) oder die Messungsorte können sich bewegen (z. B. Sportdaten). Eine beliebte Darstellung dieser Daten ist eine Karte, wahlweise als 3D-Darstellung mit der Höhe als Zeit. Dadurch sind schnelle und langsame Bewegungen unterscheidbar und Kreuzungen im Raum entsprechen Treffpunkten.

Graphen und Netzwerke Graphen und Netzwerke sind spezielle komplexe Datenstrukturen, wobei die Kanten durch zwei Keys desselben Typs auf eine Tabelle abgebildet werden können (die Values sind dann Eigenschaften der Kante, nicht der Knoten). Die Darstellung solcher Daten wird intensiv in Abschnitt 3.4 behandelt.

Bäume und Hierarchien Ein Baum kann als Spezialfall eines Graphs behandelt werden, wobei ein Value-Attribut den Key eines anderen Eintrags enthält, wodurch eine Kindheits-Relation induziert wird. Alternativ wäre eine Darstellung als Kantenliste, analog zu einem allgemeinen Graph, möglich. Im Vergleich zu allgemeinen Graphen gibt es für Bäume viele spezialisierte Visualisierungswerkzeuge, die in Abschnitt 3.4 behandelt werden. Dies geschieht durch die Ausnutzung der Ordnungsrelation, die durch einen Baum definiert wird.

2.1.2 Vorverarbeitung

Ein wichtiger Teil der Datentransformation stell die Datenvorverarbeitung dar[1]. Da reale Daten oft „unsauber" sind, müssen diese bereinigt werden, um dem „Garbage In, Garbage Out"-Prinzip zu entgehen. *Unsauber* hat dabei viele Facetten, z. B. unvollständige und fehlende Werte, Rauschen, Inkonsistenz, Ausreißer und unglaubwürdiger Werte, und viele mehr. Die Vorverarbeitung umfasst dabei alle notwendigen Transformationen, die nicht von dem*der Nutzer*in durchgeführt werden können oder sollen. Wie immer ist die genaue Abgrenzung jedoch Abhängig von der Aufgabe und dem*der Nutzer*in: Ist das Fehlen von Werten beispielsweise keine wertvolle Information für den*die Nutzer*in, so sollten diese in der Vorverarbeitung von dem*der Entwickler*in bereinigt werden. Ein anderes Beispiel sind Ausreißer oder unplausible Werte: Ist der*die Entwickler*in bei der Interpretation der Daten überfordert, sollte die Verarbeitung als Datentransformation interaktiv durch den*die Nutzer*in durchführbar sein. Der Einfachheit halber wird deshalb im folgenden immer von (Daten-) Vorverarbeitung gesprochen, wobei jeder Schritt auch als interaktive Transformation dargestellt werden könnte.

Bei der Vorverarbeitung sind insbesondere Metadaten (z. B. Attributnamen, Referenzpunkte von Messungen, Einheiten, wichtige Schlüsselwörter) hilfreich. Außerdem werden häufig statistische Methoden wie Ausreißererkennung, Clusteranalyse, Korrelationsanalyse sowie statistische Plots und Histogramme verwendet.

Fehlende Werte und Datenbereinigung

Um fehlende Werte in den Daten zu korrigieren gibt es einige grundlegende Verfahren:

- *Ignorieren:* Die fehlenden Werte werden ignoriert und es wird gehofft, dass dies in der Visualisierung nicht tiefgreifend durchschlägt.
- *Manuell Einfügen:* Unter Nutzung von Expert*innenwissen werden die Daten manuell ergänzt. Dies ist zwar präzise, aber sehr zeitaufwendig.
- *Eliminieren der Zeile:* Zeilen mit fehlenden Werten werden entfernt. Dies ist die häufigste Methode, jedoch fehlt bei vielen Datensätzen in jeder Zeile mindestens ein Wert.
- *Globale Konstante:* Anstelle der fehlenden Werte wird eine globale Konstante, z. B. -1, eingesetzt. Dies verändert allerdings die Datenverteilung, was Probleme bei statistischen Analysen hervorrufen kann.
- *Mittelwert:* Die Werte werden durch den Mittelwert ersetzt. Dies ist gut für eine globale Statistik, die individuellen Abweichungen können allerdings groß sein.
- *Wert basierenden auf Ähnlichkeit:* Ersetzung der Werte durch Annahme, welche anderen Zeilen „ähnlich" sind.

[1] Nach einer Studie von CrowdFlower im Jahr 2016 entfällt $80\,\%$ der Zeit bei der Erstellung einer Visualisierung auf die Datensuche und -vorverarbeitung und nur etwas $20\,\%$ auf die eigentliche Visualisierung.

Die konkrete Wahl einer dieser Methoden hängt dabei – wie üblich – stark von dem vorliegenden Problem ab, z. B. des Typs und der Semantik der Daten, der Menge der fehlenden Werte und der Expertise des*der Anwenders*Anwenderin. Im Allgemeinen sollte immer versucht werden, den Grund für das Fehlen zu ermitteln. Außerdem muss bei einem Ersetzen der fehlenden Werte gespeichert werden, dass es sich um Ersatzwerte handelt!

Im falle von fehlerhaften Werten, die häufig durch Menschen selbst verursacht werden, können ähnliche Methoden eingesetzt werden, allerdings sind diese sehr viel schwer zu detektieren.

Ausreißer (-detektion)

Ausreißer sind Datenpunkte, die außerhalb des normalen Wertebereichs einer Datenvariable liegen. *Starke* Ausreißer sind solche, die für Expert*innen ungewöhnlich und interessant sind. Dabei ist es nicht immer einfach zu entscheiden, was ein interessanter – aber plausibler – Ausreißer und was ein Fehler ist. Zur Detektion von Ausreißern werden deshalb häufig statistische Verfahren herangezogen. Eine einfache Methode ist die Annahme einer Normalverteilung und Ausschluss aller Werte, die um mehr als zwei Standardabweichungen vom Mittelwert abweichen. Das offensichtliche Problem dieser Annahme ist, dass sie häufig nicht gilt (bspw. da die Daten von mehreren Normalverteilungen stammen; dies kann durch Clustering detektiert werden, was später behandelt wird). Allerdings können, wie bereits erwähnt, Ausreißer auch plausibel und somit keine Datenfehler sein (ein Beispiel hierfür ist die Temperatur auf der Zugspitze, die plausibel einen Ausreißer im Vergleich zu meeresspiegelnahen Messstation darstellt).

Normalisierung und Skalierung

Oftmals würde ein einfaches Anzeigen der Daten dazu führen, dass viele der Datenpunkte an einem Ende der Skala „kleben.“ Hier kann eine Skalierung der Daten abhelfen, indem die Daten auf irgendeine Weise gestreckt/gestaucht werden. Dabei werden die Daten auf ein „normales“ Intervall, z. B. $[0, 1]$ oder $[-1, 1]$, abgebildet. Gängige Methoden sind:

- Min-Max-Normalisierung
  - Linear
  - Logarithmisch
  - Wurzel
  - Quadratisch
  - Exponentiell
- Weitere datenspezifische Normalisierungen (bspw. nach der Objektform)
- z-Score
- Quantil-Normalisierung

Bei der Min-Max-Normalisierung wird das Minimum ℓ und das Maximum u der Daten verwendet, um die restlichen Daten zu skalieren. Zur Abbildung auf das Intervall $[0, 1]$ gibt es z. B. die folgenden Methoden[2]:

$$f_{\text{Linear}}(x) = \frac{x - \ell}{u - \ell} \qquad f_{\text{Log}}(x) = \frac{\log x - \log \ell}{\log u - \log \ell} \qquad f_{\text{Quad}}(x) = \big(f_{\text{Linear}}(x)\big)^2 \qquad f_{\text{Wurzel}} = \sqrt{f_{\text{Linear}}(x)}$$

[2] Zur Interpretation der logarithmischen Normalisierung kann es hilfreich sein, diese etwas umzuformen: $(\log x - \log \ell)/(\log u - \log \ell) = \log(x/\ell)/\log(u/\ell) = \log_{u/\ell}(x/\ell) = \log_{u/\ell} x + \text{const}_x$

Für eine Normalisierung in den Bereich $[-1, 1]$, was für negative, „bipolare“, Werte sinnvoll ist, kann z. B. wieder eine Min-Max-Normalisierung

$$\hat{f}_{\text{Linear}}(x) = 2f_{\text{Linear}})(x) - 1$$

eingesetzt werden. Dabei kann es hilfreich sein, ℓ und u als die absoluten Minima und Maxima zu definieren, d. h.

$$\ell' = -\max\{|\ell|, u\} \qquad\qquad u' = \max\{|\ell|, u\},$$

um sicherzustellen, dass $\hat{f}_{\text{Linear}}(0) = 0$ gilt (dies kann sonst zu einer Verzerrung führen, bspw. bei Temperaturen).

Lokale und Globale Skalierung Gibt es mehrere Datensätze mit unterschiedlichen Minima/Maxima (z. B. Temperaturverlauf in mehreren Städten), so muss entschieden werden, welche Minima/Maxima verwendet werden. Dabei wird zwischen *lokaler* und *globaler* Normalisierung unterschieden: Bei der lokalen Normalisierung werden die Minima/Maxima jeden Datensatz bestimmt und angewendet, bei der globalen Normalisierung wird das Minimum/Maximum über alle Datensätze hinweg verwendet.

Dabei gilt, wie so oft, dass die konkrete Wahl auch von dem konkreten Problem abhängt. Während bei einer lokalen Normalisierung die Wertverläufe gut verglichen werden können (d. h. wann unterschiedliche Reihen z. B. ihr Maximum erreichen), so können dabei die Werte nicht mehr direkt verglichen werden. Komplementär dazu können die Werte bei einer globalen Normalisierung gut verglichen werden, bei großen Unterschieden ist ein Verlaufsvergleich jedoch sehr schwer.

Diskretisierung

Werden kontinuierliche Daten verarbeitet, so ist es häufig nötig, diese zu diskretisieren, d. h. diskrete Teilmengen als Approximation zu finden. Der Hauptgrund dafür ist, dass ein digitales System nicht mit „echt“-kontinuierlichen Daten umgehen kann, da sie inhärent diskret sind. Die üblichste Diskretisierungsdimension ist dabei die Zeit, bei der ein beliebiger Punkt $t_k = k\Delta_t$ für einen Zeitindex k mit vorgeschriebener Schrittweite Δ_t diskretisiert wird. Bei der Diskretisierung gehen allerdings Informationen verloren, was eventuell problematisch werden kann (vgl. Nyquist-Shannon-Abtasttheorem).

Sampling, Segmentierung und Untermengen

Um die Datenmenge zu reduzieren, gibt es verschiedene Ansätze. Eine davon ist *Sampling*, wobei zufällige Stichproben aus den Daten gezogen werden, um die Datenpunkte in der Visualisierung zu reduzieren. Je nach Wahl der Verteilungsfunktion kann sich dadurch allerdings eine Verzerrung in den Daten und einiger statistischer Eigenschaften ergeben. Andere, nicht-zufalls-basierte, Methoden sind *Segmentierung* und die Wahl von *Untermengen*. Bei der Segmentierung werden die Daten in zusammenhängende Abschnitte/Regionen eingeteilt und einer Kategorie zugeordnet. Dies ist allerdings nicht immer eindeutig möglich. Die Wahl von Untermengen kann bspw. auf Basis von Filtern und anderen Einschränkungen erfolgen.

Datenintegration

Bei der *Datenintegration* werden verschiedene Datenquellen zu einer einzigen Datenquelle vereint, was eventuell die Angleichung von Schemata und ähnlichem erfordert. Beispielsweise müssen bei der Integration imperialistischer und metrischer Quellen die Einheiten angepasst werden.

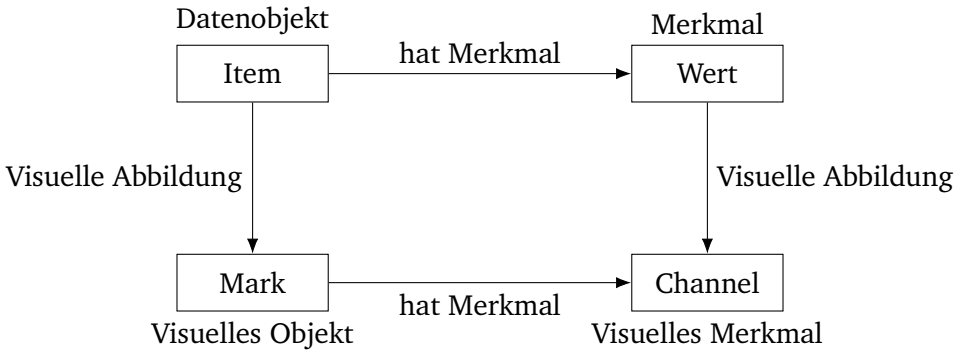

Abbildung 2.3: Beziehung zwischen Items, Werte, visuellen Channels und Marks

2.2 Die Visuelle Abbildung

In diesem Abschnitt wird die visuelle Abbildung, der dritte Schritt im Visualisierungsprozess (Abbildung 2.1) genauer behandelt. Dabei ist die Abgrenzung zum vierten Schritt, der View Transformation, nicht immer scharf gezogen, da die Schritte eng zusammenhängen. In der Regel wird in der visuellen Abbildung eine Datenvariable auf eine visuelle Struktur abgebildet. Dies muss aber nicht immer exakt sein, d. h. eine Variable kann auch auf mehrere Strukturen oder mehrere Variablen können auf die gleiche Struktur abgebildet werden (bspw. wird in einer Scatterplot-Matrix sowohl ein Datenwert als auch eine Dimension auf die Position abgebildet). Im Folgenden werden einige visuelle Strukturen („Marks“ und „Channels“) eingeführt und ihre typische Anwendung beschrieben.

Eine visuelle Abbildung muss immer die folgenden Dinge beinhalten: Titel, Skala und Achsenbeschriftungen (gegebenenfalls mit Einheit), Datenquelle, Legende und den Grafikkörper selbst. Außerdem sollte üblicherweise eine Bildunterschrift mit angegeben werden, siehe dazu **??**.

2.2.1 Beispiele

2.2.2 Visuelle Strukturen

In diesem Abschnitt werden eine visuelle Strukturen – auch *Marks* und *Channels* genannt – eingeführt und ihre übliche Einsetzung beschrieben. Dabei beginnt jede Visualisierung grundlegend mit einem leeren Blatt, dem die visuellen Strukturen hinzugefügt werden.

Der *Raum*, d. h. die Position auf der Abbildung, ist die bei weitem wichtigste und vielseitigste Struktur. Sofern nicht anders angegeben beziehen sich die Positionen immer auf zwei unabhängige[3] Variablen (x und y). Die Position ist dabei auch die einzige Struktur, die unabhängig von der Aufgabe und dem Datentyp verwendet werden kann und – ebenfalls als einzige Struktur – verwendet werden muss. Daher ist die Belegung der visuellen Raumvariablen die erste und wichtigste Designentscheidung. Innerhalb des Raumes werden anschließend weitere visuelle Strukturen platziert, die in Marks und Channels eingeteilt sind: *Marks* sind geometrische Elemente, aus denen die Visualisierung zusammengesetzt sind, während *Channels* Eigenschaften der Markierungen/Marks sind.

Die visuelle Abbildungen besteht abschließend aus einer Zuordnung von Datenvariablen und Merkmalen auf Marks und Channels, siehe Abbildung 2.3.

[3] Die Wortwahl *unabhängig* bezieht sich hier nur darauf, dass die Variablen selbst, nicht zwangsweise die Werte, unabhängig sind.

Punkt	Linie	Flächen
Position	Position	Position
Farbe	Farbe	Farbe
Helligkeit	Helligkeit	Helligkeit
Form	Breite	Kanteneigenschaften (siehe Linie)
Orientierung	Stil (Gestrichelt, Gepunktet, ...)	Tiefe
Größe	Größe/Länge	Größe/Fläche
Textur	Dynamik	Textur

Tabelle 2.1: Übliche Channels

Marks

Marks sind die eigentlichen Strukturelemente, die auf der Visualisierung abgebildet werden, die häufig jeweils ein Item oder eine Relation repräsentieren. Die üblichsten sind dabei *Punkte*, *Linien*, *Flächen* und *Text* wobei letzterer nur sehr sparsam und am besten gar nicht eingesetzt werden sollte. Zur Auswahl eines Marks sollten mehrere Aspekte in Betracht gezogen werden, insbesondere die Daten, die repräsentiert werden sollen (einzelne Items, Item-Paare, Teilmengen von Items, ein Attribut, etc.). Zum Verständnis der Visualisierung kann es hilfreich sein, wenn eine eingängige Metapher zwischen den repräsentierten Daten und der Visualisierung besteht (z. B. Darstellung eines Volumens durch eine Fläche). Dabei stellen Linien häufig dar, dass ein Änderung kontinuierlich verläuft, während Punktmengen und ähnliches andeuten, dass eine Änderung diskret ist. Des weiteren sollte bei der Auswahl der Marks darauf geachtet werden, dass die genutzten Marks die benötigten Channels (siehe nächster Abschnitt) unterstützen.

Eine Ambiguität zwischen den obigen Marks ist die Unterscheidung zwischen einem Punkt und einer Fläche: Wann ist ein Punkt „so groß“, dass er eigentlich eine Fläche ist? Dies kann dadurch aufgelöst werden, dass ein Punkt allgemein eine Markierung bezeichnet, die etwas 0-Dimensionales darstellt, d. h. auch ein großer Punkt markiert ausschließlich einen Punkt, indem der Bezug zum Hintergrund verschieden ist.

Channels

Channels sind die visuellen Eigenschaften von Marks, d. h. diese können zusätzlich als Unterscheidungs- und Identifikationsmerkmal eingesetzt werden. Dabei unterscheiden sich die anwendbaren Channels nach den eingesetzten Marks. Eine Übersicht über die üblichsten Channels ist in Tabelle 2.1 zu finden. In diesem Abschnitt werden diese üblichen Channels sowie einige nicht häufig anzutreffende Channels vorgestellt. Ebenfalls wird auf die Rolle der Elemente im Designprozess eingegangen, wobei der Grund für diese Rolle in Abschnitt 2.3 behandelt wird. Eine Übersicht über die Eignung der gängigsten Channels für verschiedene Datentypen und Aufgaben ist in Tabelle 2.2 bzw. **??** gegeben.

Im allgemeinen sollten nicht zu viele Channels verwendet werden, da die Abbildung dadurch chaotisch wird. Im konkreten Fall hängt die genaue Anzahl natürlich immer von der Aufgabe ab. Für Explain- und Explore-Aufgaben sollten generell eher weniger Channels verwendet werden, damit die Visualisierung verständlich bleibt. Außerdem sollte bei „Explore“ vermieden werden, dass dominante Channels die Wahrnehmung von Mustern in anderen Channels verhindern. Bei „Enjoy“ hingegen ist absolute Freiheit gegeben.

Farbkanäle (Farbton, Helligkeit, Sättigung) Ein sehr häufiger Channel ist die Farbe von Marks, die gleich drei verschiedene Channels aufweist: Farbton, Helligkeit und Sättigung, wobei allerdings Helligkeit und Sättigung praktisch niemals zeitgleich genutzt werden. Die Farbe kann dabei viele verschiedene Werte diskreter und

	Nominal	Ordinal	Quantitativ	Räumlich	Zeitlich
Position	+	+	+	+	+
Länge	−	+	+	?	?
Größe	−	+	○	−	−
Farbsättigung	−	+	○	−	−
Textur	+	+	−	−	−
Farbton	+	(−)	−	−	−
Orientierung	+	+	−	−	−
Form	+	−	−	−	−

Tabelle 2.2: Übliche Channels und deren Eignung für verschiedene Datentypen

	Grupp.	Sel./Hervorhebung	Vgl. Anord.	Vgl. Quant.	Unterscheidbarer Werte
Position	+	+	+	+	Bildschirmgröße
Länge	−	(+)	+	+	ca. 5 bis 15
Größe	−	+	+	+	ca. 5 bis 15
Farbsättigung	−	+	+	+	ca. 5 bis 7
Textur	+	+	+/−	+/−	ca. 5 bis 7
Farbton	+	+	−	−	ca. 7 bis 8
Orientierung	+	+	○	○	ca. 4 bis 6
Form	+	○	−	−	ca. 5 bis 7 „neutrale“

Tabelle 2.3: Übliche Channels und deren Eignung für verschiedene Aufgaben. Abkürzungen: „Grupp.“ ist „Gruppierung“, „Sel.“ ist „Selektion“, „Vgl.“ ist „Vergleich“, „Anord.“ ist „Anordnung“ und „Quant.“ ist „Quantitäten“.

kontinuierlicher Natur darstellen und ist die einzige Struktur, die auch bei nur einem Pixel funktioniert. Die Nutzbarkeit und Barrierefreiheit ist jedoch stark von der gewählten Farbskala abhängig (dies wird in Abschnitt 2.3.1 genauer behandelt).

Länge Neben der Position ist die *Länge* eines Objekts das einzige Attribut, welches numerische Größenverhältnisse exakt abbilden kann (im Falle von parallelen Längen sogar besser als die Position). Jedoch ist die Wahrnehmung stark beeinflussbar, weshalb die Darstellung sehr einfach sein muss, wenn tatsächlich numerische Vergleiche durchgeführt werden sollen, d. h. alles außer Länge und Position sollte weggelassen werden (Balkendiagramme).

Größe und Flächeninhalt Für einen qualitativen Vergleich ist es naheliegend, Größenordnungen zu vergleichen. Insbesondere bei einer Fläche sind *Differenzen* allerdings sehr schwer vergleichbar, da die Fläche quadratisch mit der Breite skaliert. Des weiteren sind Länge und Breite nicht als separate Channels zu betrachten, da sich dadurch eine Fläche ergibt, deren Größe eher wahrgenommen wird. Dadurch können die eigentlichen Daten überdeckt werden. Ein weiterer Nachteil ist, dass die Größe als Platz benötigt.

Form Die *Form* kann als Channel sehr gut gelernt werden, d. h. die Formen erhalten über die Zeit eine feste Bedeutung für den*die Nutzer*in und sie werden wiedererkannt. Da es sehr viele vorstellbare Formen gibt, könnten an sich sehr viele Formen in einer Grafik verwendet werden, jedoch ist mit üblichen Formen bereits eine Bedeutung verknüpft, was verwirrend sein kann (bspw. ein Herz als Darstellung für etwas schlechtes oder Pfeile, die auf nichts zeigen). Daher werden in der Praxis häufig „neutrale“ Formen wie Kreise, Quadrate, oder Dreiecke verwendet.

Orientierung Die *Orientierung* eines Marks kann ebenfalls genutzt werden, um Daten darzustellen (bspw. die Orientierung eines Pfeils). Natürlich kann dies nur bei bestimmten Formen genutzt werden, abhängig von den Symmetrien der Form. Außerdem ist der Übergang von einer Form mit variabler Orientierung und zwei verschiedenen ähnlichen Formen nicht strikt.

Exoten Einige Exoten unter den Channels sind bspw. Tiefe/Überdeckung, Schatten (und die Richtung dessen), Textur, Animation/Bewegung, Schattierung und Unschärfe. Diese sollen allerdings nicht oder nur sehr sparsam eingesetzt werden!

Glyphen

Glyphen sind eine spezielle Art von Marks, die insbesondere bei kartographischen Abbildungen (siehe Abschnitt 3.5) verwendet werden. Ein Glyph kann dabei beliebig komplex sein und viele Channels und Marks kombinieren, sollten dabei aber intuitiv bleiben; sie stellen eine *gemischte Visualisierung* dar. Bei Seekarten sind dies zum Beispiel die Markierungen der Tonnen auf den Seestraßen (siehe Abbildung 2.4). Formal ist ein Glyph ein „kleines, unabhängiges, visuelles Objekt, welches Merkmale von einem oder mehreren Items zeigt“. Dabei kann ein Glyph viele andere Zeichen und Visualisierungen umfassen. Da diese Definition sehr allgemein ist, muss bei dem Design von Glyphen immer zwischen Komplexität eines Glyphs und der zu erwartenden Anzahl Glyphen abgewägt werden. Dabei sollten die Glyphen einfacher sein, wenn Muster statt Werte relevant sind oder die Wertverteilung „chaotisch“ sein könnte. Wie auch bei „großen“ Visualisierungen ist es wichtig festzulegen, welche Datenattribute welche Channels nutzen (hier können alle bereits erwähnten und noch zu erwähnenden Richtlinien angewandt werden).

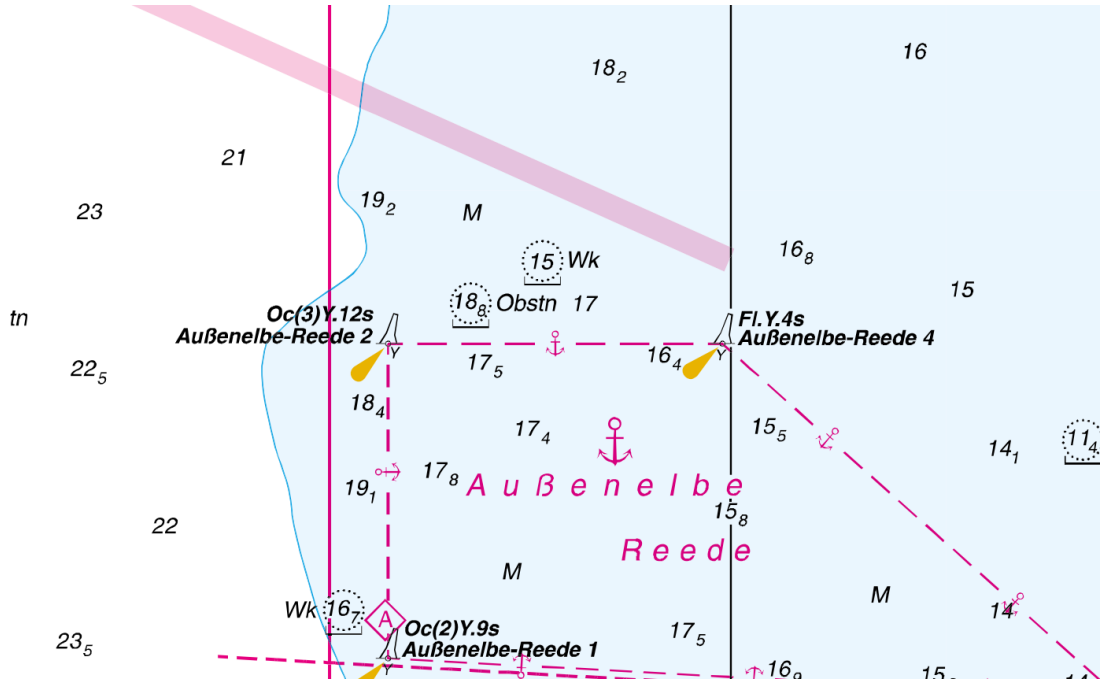

Abbildung 2.4: Auf diesem Seekartenausschnitt ist zu sehen, wie die Tonnen Außenelbe-Reede 2 und 4 durch Glyphen dargestellt werden, die viele Informationen transportieren. Bei ersterer ist bspw. die Information enthalten, dass es sich um eine gelbe einfarbige Tonne ohne Toppzeichen mit unterbrochenem Feuer in dreier-Gruppen in Gelb und einer Wiederkehr von $12\,\mathrm{s}$ handelt.

Als noch weiter gefasste Variante eines Glyphs kann ein solches sogar selbst wieder Visualisierungen enthalten, bspw. Starplots innerhalb eines Scatterplots. Häufig genutzt werden außerdem Torten-, Linien- und Balkendiagramme. Dabei nutzen die Glyphen nur den Grafikkörper, Achsen und Achsenbeschriftungen werden bewusst ausgelassen, um die Darstellung zu komprimieren.

2.2.3 Bildunterschriften

2.3 Wahrnehmung, Position und Layout

2.3.1 Wahrnehmungsmodelle von Ware

Farbe und Farbmodelle

Color-Mapping

2.3.2 Elementare Visuelle Aufgaben

Anwendersicht

Beispiel

Suche

Queries

2.3.3 Eigenschaften Verschiedener Visueller Channels

Auswahl/Hervorhebung

Ordnung

Differenzen

Zusammenfassen

2.3.4 (Ungewollte) Einflüsse

Kontrast und Perzeptuelle Länge

Farbnamen und Farbkategorien

Konsistente Bewegung und 3D aus Bildern

Kontrast und Kontext

Separierende und Integrierende Channels

2.3.5 Position, Layout und Komposition

Beispiele

Zusammengesetzte Visualisierungen

Beispiele

2.4 Interaktion

2.4.1 Benutzungsschnittstellen

Bedienung und Interaktion nach Norman

ISO 9241

Interaktionsmodi nach Spence

Kontinuierliche Interaktion

Schrittweise Interaktion

Passive Interaktion

Gemischte Interaktion

2.4.2 Interaktionstechniken

Systemnahe Interaktionstechniken

Selektion

Navigation

Shneidermans Mantra

Fokus und Kontext

Überblick und Detail

Brushing und Linking

Kategorien der Interaktion nach Yi et al.

2.4.3 Design

Leitsätze

Navigation

Organisation

Erzeugung von Aufmerksamkeit

Unterstützung der Dateneingabe

Prinzipien

Ermittlung des fachlichen Niveaus des Nutzers

Ermittlung der Arbeitsaufgaben

Wahl des Interaktionsstils

Menschliche Reaktionszeit

3 Spezialisierte Visualisierungstechniken

3.1 Hochdimensionale Daten

3.1.1 Quantitative Daten

Scatterplot-Matrix

Starplot

Small Multiple Plots

Parallele Koordinaten

. . . mit Interaktion

RadViz

3.1.2 Kategorische Daten

Parallel Sets

. . . mit Interaktion

Mosaic-Plot

KV-Map

Tabelle

3.2 Große Datenmengen

3.2.1 Ordnen

Dimensionsreduktion und Feature Selektion

Principal Component Analysis (PCA)

Linear Discriminant Analysis (LDA)

Multidimensional Scaling (MDS)

Self-Organizing Map (SOM)

3.2.2 Aggregieren

3.3 Zeitbasierte Daten

3.3.1 Viele Zeitreihen

Filtern

Heatmaps

. . . mit Aggregation

Horizon Plots

Small Multiples

. . . mit Aggregation

3.3.2 Periodische Zeitreihen

Spiral Layouts

Matrix-Layout

3.3.3 Diskrete Ereignisse

Sequenzbaum

3.4 Graphen und Bäume

3.4.1 Bäume

Node-Link-Diagramm

Radiales Layout

TreeMaps

Cushions

Squarified

Icicle Plot und Sunburst

3.4.2 Allgemeine Graphen

Layouts

Force-Directed

Layer-Based: Sugiyama

Constraint-Based: Metro-Map

(Hierarchisches) Edge-Bundling

3.4.3 „Search, Show Context, Expand on Demand“

3.5 Geobasierte Daten und Karten

3.5.1 Karten als Metapher

Karten und Schematisierungen

3.5.2 Geobezogene Daten

Kartenprojektion

Plattkarte

Mercator Projektion

Winkel-Tripel

Verzerrte Darstellungen

Metro-Map

(Stetige) Kartogramme

Abstrakte Geovisualisierungen

3.5.3 Nicht-Geobezogene Daten

Wikipedia World Map

Themescapes

Themengebiete

Gmap World of Music

Metro-Map Immitation

Rekonstruktion von Terrain aus Knotenattribut

3.5.4 Raum-Zeit Daten

Darstellung von Richtungen

Darstellung von Geschwindigkeiten

Darstellung von Vielen Trajektorien