

Universität Konstanz
Fachbereich Informatik und Informationswissenschaft
Bachelor-Studiengang Information Engineering

Bachelorarbeit

Konzeption und Entwicklung einer Visualisierung für mehrdimensionale Daten

zur Erlangung des akademischen Grades eines
Bachelor of Science (B.Sc.)

Studienfach: Information Engineering
Schwerpunkt: Computer Science
Themengebiet: Angewandte Informatik

von

Mischa Demarmels

(01/562779)

Erstgutachter: Prof. Dr. Harald Reiterer
Zweitgutachter: Prof. Dr. Oliver Deussen
Betreuer: Prof. Dr. Harald Reiterer
Einreichung: 6. März 2007

Inhaltsverzeichnis

1	Einleitung	6
1.1	Motivation	6
1.2	Evolution von Scatterplots	7
1.3	Inhaltlicher Aufbau	9
2	Definitionen	11
2.1	Zoomable User Interface	11
2.2	Geometrischer Zoom	11
2.3	Semantischer Zoom	12
2.4	Geometrisch-semantischer Zoom	13
2.5	Scatterplot	14
2.6	Datapoint Cluster	14
2.7	Multiple Datapoint	15
3	Projekt MedioVis	17
3.1	Historische Entwicklung von MedioVis	17
3.2	Ziele bei der Entwicklung von MedioVis	19
3.3	Anwendungsdomäne	20
3.4	Definition des Datenraumes	21

<i>INHALTSVERZEICHNIS</i>	3
3.5 Definition des Use-Cases für diese Arbeit	23
4 Klassische Benutzung von MedioVis	24
4.1 Anwendungsszenario Teil I (HyperGrid und ScatterPlot)	24
4.2 Die HyperGrid	28
4.2.1 Aspect of Interest	29
4.2.2 Degree of Interest und Level of Detail	30
4.2.3 HyperGrid-Browser	32
4.2.4 Spaltenfilter und Sortierung	35
4.2.5 Extraktion von Detailinformationen aus der HyperGrid	36
4.3 Der ScatterPlot	38
5 HyperScatter	42
5.1 Ziele des HyperScatters	42
5.1.1 Gründe für eine Scatterplot-Visualisierung	42
5.1.2 Gründe für ein Zoomable User Interface	44
5.2 Interaktionskomponenten des HyperScatters	45
5.3 Visualisierungs- und Interaktionskonzept des HyperScatters	46
5.3.1 Overview	46
5.3.1.1 Übersichtsdarstellung mit dem HyperScatter	47

5.3.1.2	Aktualisierung der beiden Achsen und der Legende des HyperScatters	49
5.3.1.3	Technische Umsetzung der Verortung von Datenpunkten im HyperScatter	50
5.3.2	Relate	52
5.3.2.1	Visualisierung von Zusammenhängen mit dem HyperScatter	52
5.3.2.2	Medientyp-Symbole als dritte Dimension	53
5.3.2.3	Variable Achsenbelegung im HyperScatter	53
5.3.3	Details-On-Demand	54
5.3.3.1	Geometrischer vs. semantischer Zoom	55
5.3.3.2	Detail-Zoom im HyperScatter	56
5.3.3.3	Detail-Zoom von Single Datapoints	58
5.3.3.4	Detail-Zoom von Multiple Datapoints	61
5.3.3.5	Browser-Komponente im HyperScatter	64
5.3.3.6	Technische Umsetzung des animierten Zoomings von Java-Components im HyperScatter	65
5.3.4	Zoom	66
5.3.4.1	Geometrisches Zooming unter Verwendung einer Bounding-Box	67
5.3.5	Filter	69
5.3.5.1	Globales Filtersystem in MedioVis	70

<i>INHALTSVERZEICHNIS</i>	5
5.3.6 Extract	71
5.3.6.1 Selektion einzelner Datenobjekte	72
5.3.6.2 Selektion mehrerer Datenobjekte	72
5.3.6.3 Weiterverarbeitung von Selektierten Datenobjekten	74
5.3.7 History	76
5.3.7.1 Suchanfragen-History	76
5.3.7.2 Interaktions-History	77
5.4 Anwendungsszenario Teil II (HyperScatter)	78
6 Integration des HyperScatters in den MedioVis-Client	80
6.1 Integration durch das Tabbed-Document-Interface-Konzept	80
6.2 Koordination der Selektion von Datenobjekten	81
6.3 Konsistenz der Interaktions- und Visualisierungskonzepte	82
7 Zusammenfassung und Ausblick	83

1 Einleitung

1.1 Motivation

Das Internet ist in der heutigen Zeit zu einer der wichtigsten und meistgenutzten Informationsquellen geworden. Dabei zeichnet es sich vor allem durch seinen umfassenden Datenbestand aus. Um eine sinnvolle Nutzung dieser Datenfülle zu ermöglichen, müssen leistungsstarke Portale für die Recherche im Internet bereitgestellt werden. Diese Aufgabe wird von Web-Suchmaschinen wie beispielsweise Google¹, Yahoo² und MSN³ geleistet, welche zusammen diesen Markt dominieren.

Obwohl die heute verbreiteten Internet-Technologien der Informationsvisualisierung kaum noch Restriktionen aufzwingen, wird bei den marktführenden Web-Suchmaschinen weiterhin an den herkömmlichen Konzepten, der listenbasierten Darstellungen von Treffermengen, festgehalten.

Dabei kann gezeigt werden, dass moderne *visuelle Suchsysteme*, wie sie zum Beispiel von Shneiderman schon 1996 durch sein Visual Information Seeking Mantra [Shn96] charakterisiert werden, den listenbasierten Trefferdarstellungen von Web-Suchmaschinen in weiten Bereichen überlegen sein können [Ger04]. Trotzdem stossen neue Ansätze von (auch interaktiven) Informationsvisualisierungen (wie beispielsweise des Web-Suchdienstes KartOO⁴) heute auf wenig Akzeptanz und können sich kaum durchsetzen. Die beschriebene Problematik beschränkt sich keineswegs nur auf die Websuche. Auch in anderen Domänen wie Katalogsuchen, Online-Hilfesystemen, dem Dokumentenmanagement oder der Suche in grossen Web-Foren treten die gleichen Symptome auf.

In dieser Arbeit soll deshalb der Ansatz verfolgt werden, bei der Konzeption und Entwicklung eines visuellen Suchsystems der nächsten Generation, neben dem Entwickeln von neuen inovativen Interaktions- und Visualisierungskonzepten, gleichzeitig auf ein altbewährtes und akzeptiertes Konzept zu setzen: Dem Scatterplot.

¹<http://google.com/>

²<http://yahoo.com/>

³<http://www.msn.com/>

⁴<http://www.kartoo.com/>

1.2 Evolution von Scatterplots

Im Hauptteil dieser Arbeit wird eine Visualisierung namens HyperScatter vorgestellt, die neben dem Zoomable User Interface das Konzept des Scatterplots umsetzt. Um die Entscheidung für dieses Konzept zu motivieren, soll im Folgenden schemenhaft anhand von drei evolutionären Beispielen von Scatterplot-Visualisierungen einige Vorteile dieser Art der Informationsvisualisierung aufgezeigt werden.

Scatterplots haben eine lange Tradition in der Informationsvisualisierung. Dabei wurden sie schon lange vor der Computer-Ära zur Darstellung von Zusammenhängen meist quantitativer Daten eingesetzt. Die genauen Wurzeln der Scatterplot-Visualisierung sind kaum mehr nachvollziehbar. Eine der frühesten Abbildungen eines Scatterplots, welcher aus der Beobachtung von Barometer-Werten in Relation zur Höhenlage abgeleitet wurde, stammt aus dem Jahr 1686 von Edmund Halley (siehe Abbildung 1).

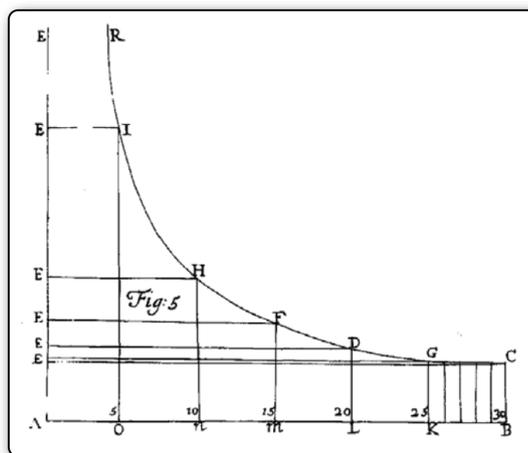


Abbildung 1: Scatterplot von Edmund Halley 1686, Darstellung von Barometer-Werten in Relation zur Höhenlage. *Quelle: [FD05]*

Auch in unserer Zeit haben Scatterplots eine grosse Bedeutung bei der Informationsvisualisierung. Edward Tufte schätzte 1983, dass zwischen 70 und 80 Prozent aller Diagramme, die in wissenschaftlichen Arbeiten publiziert werden, Scatterplots sind [Tuf83]. Die Vorteile von Scatterplots gegenüber textuellen, listenbasierten oder tabellarischen Visualisierungen liegen vor allem darin, dass sie das Erkennen von Zusammenhängen und Trends durch die visuelle Aufbereitung der Datengrundlage vereinfachen. In Abbildung 2 können zum Beispiel aus den beiden oberen Scatterplots problemlos Trends der Datenverteilung

herausgelesen werden. Bei den unteren beiden Diagrammen ist der jeweilige Ausreisser ebenfalls auf den ersten Blick identifizierbar. Dies ist anhand der tabellarischen Darstellung derselben Daten (jeweils links neben dem Scatterplot) nur nach intensiver Analyse der Zahlenwerte möglich.

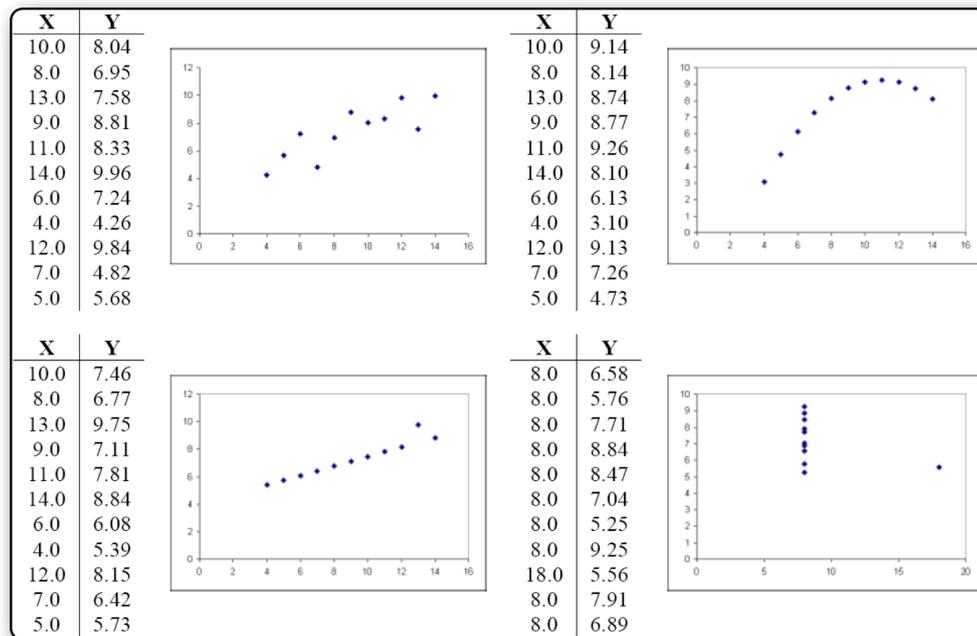


Abbildung 2: „The power of visual representations“ Graphik von Edward Tufte, modifiziert von Barry Wilkins *Quelle: [Wil03]*

Mit dem Einzug des Computers in unseren Alltag, konnte sich neben dem Information Retrieval und dem Data Mining auch die Informationsvisualisierung wesentlich weiterentwickeln. Einen entscheidenden Vorteil stellt dabei die vom Medium Computer ermöglichte Interaktivität von Informations- und Wissensrepräsentationen dar [ERST05].

In diesem Zusammenhang sind auch interaktive Scatterplot-Visualisierungen entstanden. Der FilmFinder [AS94b] ist eines der bekanntesten Beispiele dafür. Dieser visualisiert in einem Scatterplot den Inhalt einer Filmdatenbank (siehe Abbildung 3). Dabei wird jeder Film als verschiedenfarbiger Punkt dargestellt. Die Verortung auf dem Scatterplot geschieht über die beiden zu den Filmen abgespeicherten Metadaten Erscheinungsjahr und Popularität. Ein weiteres Attribut (das Genre) ist über die Farbe der Datenpunkte codiert. Die restlichen Metadaten der Filme sind über ein Pop-Up-Fenster zugänglich, welches erscheint, wenn ein Benutzer einen Datenpunkt anklickt. Neben seiner Eigenschaft

als Scatterplot, verwirklicht der FilmFinder zusätzlich das Konzept von Dynamic Queries [AWS92]. Über diverse Filter-Widgets kann die Darstellung des gesamten Datenraumes auf die für den Benutzer relevanten Filme eingeschränkt werden.

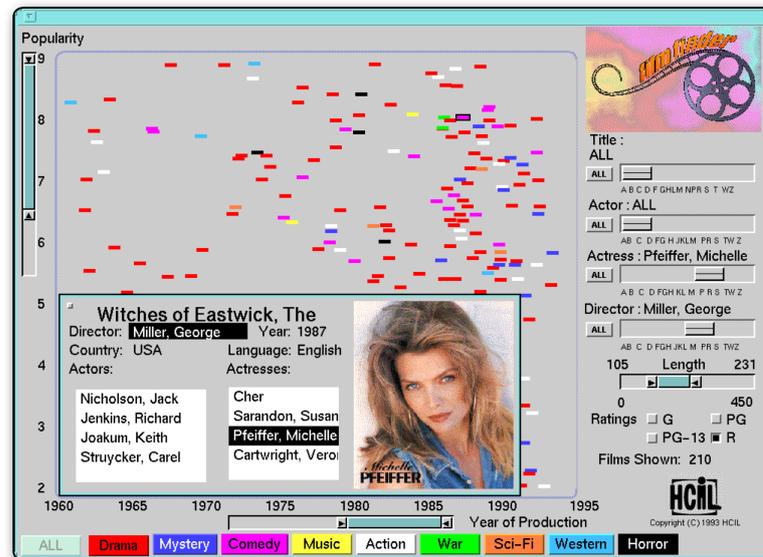


Abbildung 3: Der FilmFinder von Ahlberg und Shneiderman.

Diese drei Beispiele sollen aufzeigen, dass Scatterplot-Visualisierungen schon seit langem und auch heute noch bei der Informationsvisualisierung eine wichtige Rolle einnehmen. Dabei können sie, wenn sie mit geeigneten Interaktionskonzepten kombiniert werden auch als zukunftsweisende interaktive Werkzeuge zur Datenanalyse im Rahmen von visuellen Suchsystemen eingesetzt werden.

1.3 Inhaltlicher Aufbau

Nachdem im anschliessenden Kapitel 2 einige kurze Gebrauchsdefinitionen einzelner, in dieser Arbeit verwendeter, Begriffe gegeben werden, wird in Kapitel 3 das Projekt MedioVis vorgestellt. Dabei wird neben der historischen Entwicklung vorallem die primäre Anwendungsdomäne von MedioVis definiert, mit welcher auch im Hauptteil dieser Arbeit argumentiert wird.

Nach einem konkreten Anwendungsszenario, werden im Kapitel 4 zwei Visualisierungen vorgestellt (die HyperGrid und der ScatterPlot), anhand derer bereits einige Konzepte

erläutert werden, welche auch für die im Hauptteil dieser Arbeit vorgestellten Visualisierung — dem HyperScatter — von Bedeutung sind. Dieser soll dann in Kapitel 5 anhand des Visual Information Seeking Mantras von Ben Shneiderman [Shn96] im Detail vorgestellt werden. Im Kapitel 6 soll darauf eine Möglichkeit aufgezeigt werden wie die entwickelte Visualisierung in das MedioVis-Framework integriert werden kann.

Abgeschlossen wird diese Arbeit schliesslich mit einer Zusammenfassung und einem kurzen Ausblick auf mögliche Anknüpfungspunkte für weitere Projekte im Zusammenhang mit der Weiterentwicklung von MedioVis und dem HyperScatter.

2 Definitionen

2.1 Zoomable User Interface

Ein *Zoomable User Interface* ist eine spezielle Ausprägung einer grafischen Benutzeroberfläche. Es bietet dem Benutzer die Möglichkeit, in die Programmoberfläche hinein, oder aber nur einzelne Elemente der Applikation heran zu zoomen. Dabei ist es dem Benutzer meistens möglich, den Skalierungsfaktor des Zooms interaktiv zu steuern.

Neben dem *Zooming* ist auch das *Panning* eine wichtige Interaktion mit einem Zoomable User Interface [HBP02]. Hierbei wird dem Benutzer die Möglichkeit geboten, die gezoomte Oberfläche horizontal und vertikal zu verschieben. Somit kann der Anwender Bereiche, die nicht in der Grenze der angezeigten Fläche liegen, in den sichtbaren Bereich der Anwendung rücken.

Eine State-of-the-Art-Analyse zu Zoomable User Interfaces ist zum Beispiel hier [Kön06] zu finden.

2.2 Geometrischer Zoom

Das *geometrische Zooming* kann in zwei unterschiedliche Ausprägungen eingeteilt werden: Einerseits wird von einem geometrischen Zoom gesprochen, wenn von einer Übersichts-Darstellung immer näher an einen bestimmten Punkt herangezoomt wird (siehe Abbildung 4, oben). Andererseits gehört auch das Heranzoomen eines einzelnen Objekts an den Betrachter in die Kategorie des geometrischen Zoomings (siehe Abbildung 4, unten).

Die oben als Erstes beschriebene Art des Zoomings begegnet einem ständig im Alltag, wenn man sich einem Objekt (zum Beispiel einer Anzeigetafel) annähert, um die darauf repräsentierten Informationen besser erkennen zu können. Auch entspricht der Kamera-zoom in der Filmwelt dieser Ausprägung des Zoomings.

Die zweite angesprochene Art des geometrischen Zoomings verhält sich dazu sehr ähnlich, nur bleibt hier der umliegende Kontext der gleiche. Das heisst, es verändert nur ein einzelnes Objekt seine Grösse. Dieses verdeckt dabei je nach Zoomrichtung immer mehr oder

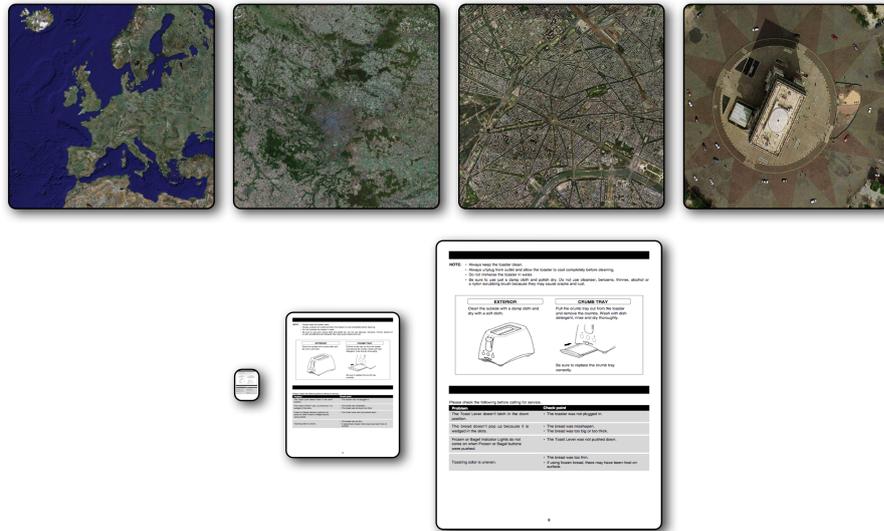


Abbildung 4: Geometrischer Zoom in ein Satellitenbild (oben), geometrisches Heranzoomen eines PDF-Dokumentes (unten).

weniger des umliegenden Kontexts. Dieser bleibt aber grundsätzlich bestehen und auch seine Größe bleibt erhalten.

2.3 Semantischer Zoom

Beim semantischen Zooming handelt es sich um das Konzept, ein Objekt, je nach dessen semantischer Zoomstufe, mehr oder weniger Informationsgehalt darstellen zu lassen. Manchmal wird nicht nur die Menge an dargestellten Informationen verändert, sondern es werden je nach semantischer Zoomstufe, Informationen ausgetauscht.



Abbildung 5: Semantischer Zoom, dargestellt anhand der Daten eines Filmdatensatzes.

Abbildung 5 versucht einen semantischen Zoom zu symbolisieren. Wichtig ist dabei, dass das semantische Zoomen an sich keine Auswirkung auf die Grösse eines dargestellten Objekts hat.

2.4 Geometrisch-semantischer Zoom

Oft werden die beiden Zooming-Verfahren kombiniert eingesetzt, indem zum Beispiel je nach Skalierung eines gezoomten Objektes zusätzlich an den zur Verfügung stehenden Platz angepasste Meta-Daten eingeblendet werden [PF93]. In Abbildung 6 wird als Beispiel ein PDF-Dokument herangezoomt. Durch das geometrische Zooming nimmt die visuelle Repräsentation des PDF-Files immer mehr Platz ein. Der so zur Verfügung stehende Platz wird je nach Grösse mit mehr Meta-Daten angereichert, was einem semantischen Zoom entspricht. Sobald das Dokument gross genug ist um den wirklichen Inhalt erkennen zu können, werden die Meta-Daten wieder ausgeblendet.

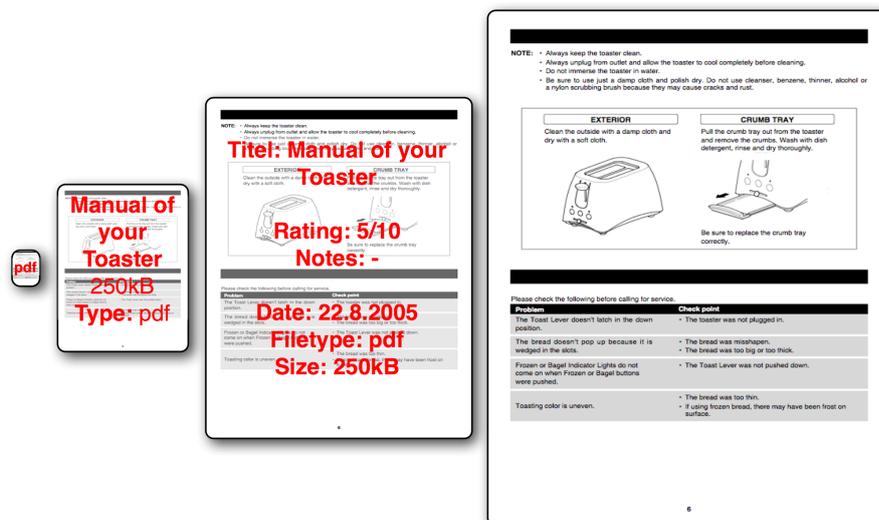


Abbildung 6: Geometrisch-semantischer Zoom, dargestellt anhand eines PDF-Dokuments.

Meistens wird diese Art des kombinierten geometrisch-semantischen Zoomings schlicht als semantisches Zooming referenziert.

2.5 Scatterplot

Ein Scatterplot (auch Punkt- oder Streudiagramm) ist eine Visualisierung, mit der im Prinzip beliebig viele Datenpunkte auf einer vorgegebenen Fläche dargestellt werden können. Einschränkungen werden in den folgenden Abschnitten *Datapoint Cluster* und *Multiple Datapoint* dargelegt. Jeder Datenpunkt wird anhand zweier oder mehrerer Achsen in einem Koordinatensystem angeordnet (siehe Abbildung 7, links). Am häufigsten und am einfachsten darzustellen sind zweidimensionale Scatterplots, wobei sich aber auch Beispiele für dreidimensionale Scatterplots finden lassen [Kön03].

Damit sich ein Datenbestand dazu qualifiziert in einem Scatterplot visualisiert werden zu können, müssen die Datenpunkte mindestens so viele Attribute aufweisen, wie der Scatterplot Dimensionen hat. Durch die Belegung der Achsen des Diagramms werden dann alle Datenpunkte, entsprechend ihrer Ausprägungen in diesen Dimensionen, angeordnet. So lassen sich leicht Relationen der Datenpunkte im Datenbestand ersehen.

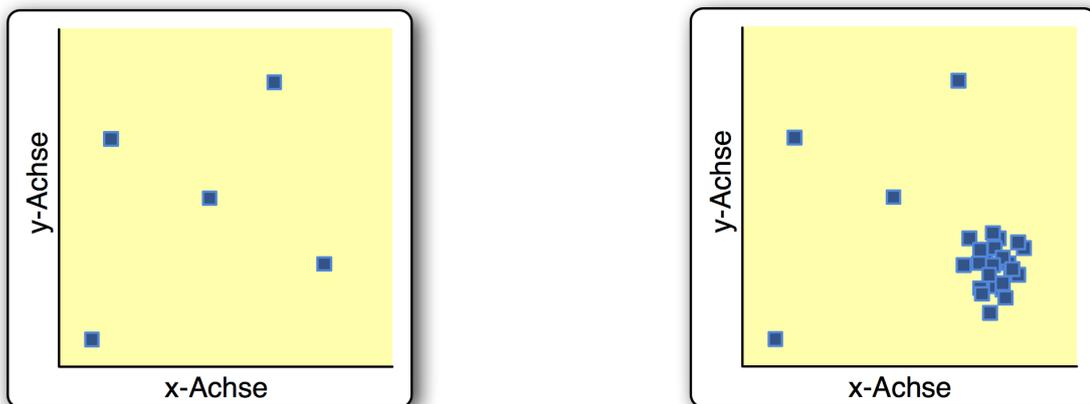


Abbildung 7: Ein zweidimensionaler Scatterplot (links), ein Scatterplot mit einer Clusterbildung (rechts).

2.6 Datapoint Cluster

Unter Datapoint Clustern versteht man, wenn in einem Scatterplot die einzelnen Datenpunkte in gewissen Bereichen des Diagramms so nahe beieinander liegen, dass diese anfangen sich gegenseitig zu überlagern. Dieses Phänomen tritt vor allem dann auf, wenn sich im Datenbestand viele Datenpunkte mit ganz ähnlichen Ausprägungen in den auf die

Achsen des Scatterplots abgetragenen Dimensionen befinden (siehe Abbildung 7, rechts). Die Wahrscheinlichkeit des Auftretens von Datapoint Clustern erhöht sich im Allgemeinen mit der Anzahl angezeigter Datenpunkte.

Einerseits kann in diesem Fall beim Betrachten der Visualisierung festgestellt werden, dass in der Region der Clusterung offensichtlich ein erhöhtes Aufkommen von Datenpunkten herrscht. Es kann also eine Aussage über einen gewissen Wertebereich des Scatterplots getroffen werden. Andererseits ist es in dieser Region aber unter Umständen nicht mehr möglich, die einzelnen Datenpunkte voneinander getrennt wahrzunehmen. Weiter hinten liegende (also früher gezeichnete) Datenpunkte können sogar von weiter vorne liegenden Punkten vollkommen überdeckt werden. Dies kann es verunmöglichen Aussagen über diese konkreten Datenpunkte zu treffen.

2.7 Multiple Datapoint

Wenn mehrere Datenpunkte in den auf die Achsen abgetragenen Dimensionen eines Scatterplots genau die gleichen Ausprägungen aufweisen, kommen sie im Diagramm exakt an der gleichen Stelle zu liegen. Dadurch ist es nicht möglich, alle diese Datenpunkte darzustellen, sondern es ist nur der zuletzt gezeichnete Datenpunkt sichtbar.

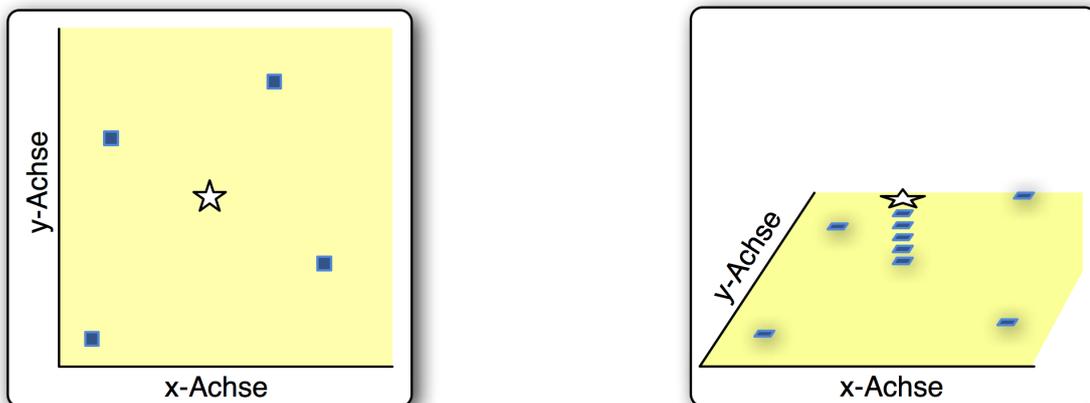


Abbildung 8: Links: Scatterplot mit einem durch einen Stern gekennzeichneten Multiple Datapoint. Rechts: Schematische Darstellung, wie der Multiple Datapoint zustande kommt und was er repräsentiert.

Um diesem Problem zu begegnen, wird ein virtueller Datenpunkt (welcher Multiple Datapoint genannt wird) geschaffen, welcher mehrere normale Datenpunkte beinhalten kann (siehe Abbildung 8). Dieser wird statt der richtigen Datenpunkte an deren Stelle im Scatterplot gezeichnet. Damit zwischen normalen Datenpunkten und Multiple Datapoints unterschieden werden kann, werden letztere durch eine andere visuelle Repräsentation im Diagramm ausgezeichnet (im Beispiel von Abbildung 8 durch einen Stern).

3 Projekt MedioVis

MedioVis ist ein Forschungsprojekt der *Arbeitsgruppe Human-Computer Interaction*⁵ der Universität Konstanz und stellt ein visuelles Suchsystem der nächsten Generation dar. Die Konzepte, welche in dieser Arbeit vorgestellt werden stehen in starkem Bezug zu dem Projekt, welches auch die Grundlage zum verwendeten Use-Case in dieser Arbeit liefert. Aus diesem Grund soll das Projekt, beginnend mit der historischen Entwicklung von MedioVis, vorgestellt werden.

3.1 Historische Entwicklung von MedioVis

MedioVis baut auf den früheren Forschungsprojekten der Arbeitsgruppe Human-Computer Interaction auf und kann damit von langjährig gewonnenen Erfahrungen profitieren. In historischer Reihenfolge sind dies INSYDER⁶, INVISIP⁷ und VisMeB⁸.

Das 1998 gestartete Projekt INSYDER [RMMH00] war eine Anwendung zur Suche und visuellen Analyse von aus wirtschaftlicher Sicht relevanten Informationen aus dem Internet. Schon damals wurde bei der Entwicklung teilweise auf die Programmiersprache Java⁹ gesetzt, welche sich bis heute, und somit auch in MedioVis, als solide Grundlage zur Implementation und zur Distribution der Prototypen und Anwendungen ausgewiesen hat.

Ende 2001 folgte das Projekt INVISIP. Dabei handelte es sich um einen Metadaten Browser zur Suche und Analyse von geographischen Daten. Ziel war es, eine Anwendung zu schaffen, welche bei Standortentscheidungen die partizipierenden Parteien in der Planung und der Durchführung begleitet und unterstützt. Um den wechselnden Einsatzumgebungen als Applikation oder als Applet und den gestiegenen Anforderungen an die Datenanbindung (XML-Files und Online-Datenbanken) bei INVISIP gerecht zu werden, wurde das auf dem INSYDER-Projekt aufbauende Framework einem ersten Redesign unterzogen. Auch kamen weitere Visualisierungen hinzu, für welche ein geeignetes Framework geschaffen werden musste, damit sie untereinander und mit dem Datenmodell synchronisiert werden konnten.

⁵<http://hci.uni-konstanz.de/>

⁶<http://www.insyder.com/>

⁷<http://www.invisip.de/>

⁸<http://hci.uni-konstanz.de/index.php?a=research&b=projects&c=3042480&lang=en>

⁹Java, Sun Microsystem, <http://java.sun.com/>

Auf Grund der stetig wachsenden Anforderungen wurde es schliesslich nötig ein noch flexibleres Framework zu programmieren. Dies sollte mittels Generalisierung und Abstraktion von bisherigen Konzepten und der Modularisierung der verschiedenen bereits entwickelten Visualisierungen erreicht werden. Das Resultat dieses Vorhabens wurde unter dem Namen VisMeB (Visueller Metadaten Browser) [KRML03] veröffentlicht. Bei VisMeB war es nun möglich ganz verschiedene Visualisierungen, wie GranularityTable und LevelTable [LRKM03], CircleSegmentView [KR05], ScatterPlot und 3D-ScatterPlot [Kön03], auf einfache Weise in die Applikation zu integrieren. Dabei konnten die verschiedenen Visualisierungen je nach Anforderung reorganisiert und neu kombiniert werden (siehe Abbildung 9).

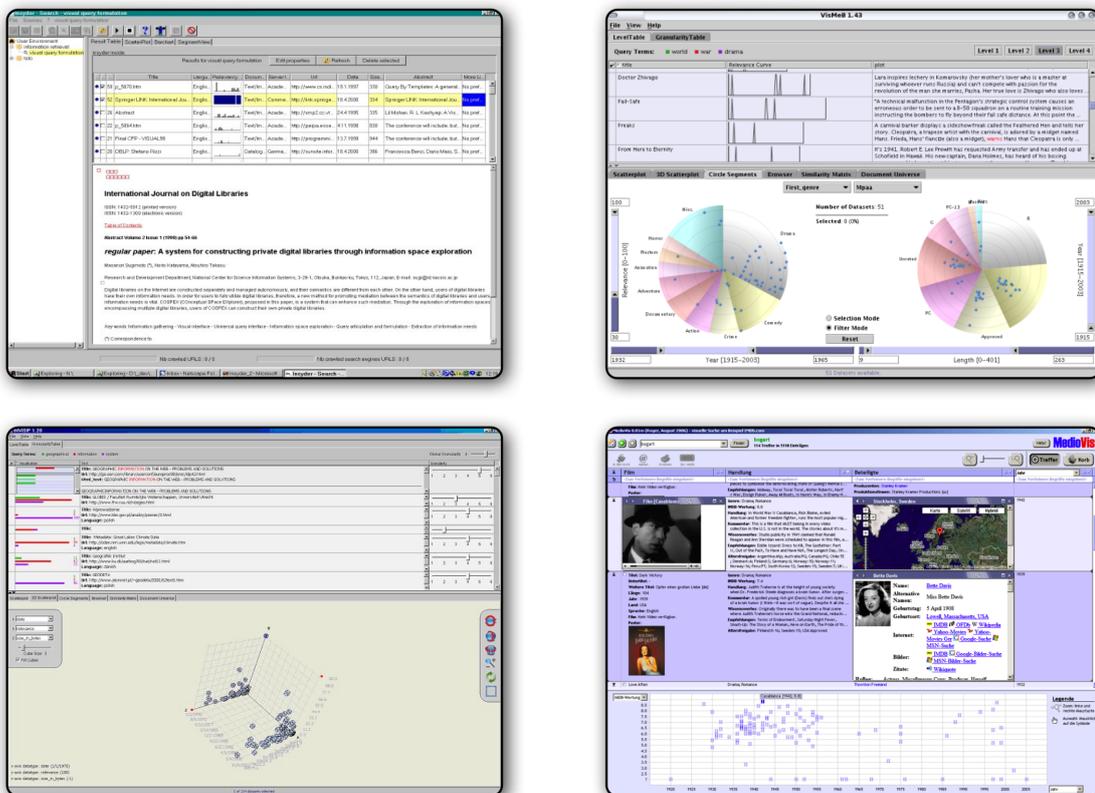


Abbildung 9: Evolution der Suchsysteme am Lehrstuhl HCI (von oben links nach unten rechts): INSYDER; VisMeB mit LevelTable und CircleSegmentView; VisMeB mit GranularityTable und 3D ScatterPlot; MedioVis mit HyperGrid und ScatterPlot.

Im Jahr 2004 konnte schliesslich durch die Partnerschaft mit der *Bibliothek der Universität Konstanz*¹⁰ eine neue Einsatzdomäne für das Projekt gefunden werden. Diese war aus vielerlei Gründen sehr attraktiv für das wissenschaftliche Arbeiten in der Arbeitsgruppe: Einmal waren die Katalogdaten der Bibliothek durch ihre starke Heterogenität und Multidimensionalität und auch durch die Grösse des Datenbestands eine sehr interessante Datengrundlage um neue Konzepte auszuprobieren und diese auch auf deren Skalierbarkeit testen zu können. Dazu kam, dass durch die geographische Nähe der *Bibliothek der Universität Konstanz* ein schneller Austausch zwischen der Arbeitsgruppe und den Bibliotheksverantwortlichen stattfinden konnte. Auch erleichterte die Kooperation das Evaluieren der nun unter dem Namen MedioVis im Live-Betrieb in der Mediothek, einem Teilbereich der Bibliothek, zum Einsatz kommenden Anwendung. Dort konnten nun Langzeit-Evaluationen mit „echten“ Benutzern durchgeführt werden. Priorität wurde dabei auch auf den Schritt weg von einem Expertensystem (wie noch in VisMeB) und hin zu einem leicht zu verstehenden System, das jedermann, auch unerfahrene Nutzer, bedienen kann, gelegt. Auch MedioVis durchlebte seither einige Veränderungen am Grundsystem und wird auch aktuell immer wieder an die neuen Bedürfnissen bezüglich Flexibilität und Funktionalität angepasst.

3.2 Ziele bei der Entwicklung von MedioVis

Im Projekt MedioVis geht es nicht einzig darum, eine Anwendung im Sinne eines visuellen Suchsystems zu entwickeln. MedioVis soll auch ein Framework bieten, welches das bequeme Entwickeln von Visualisierungs- und Interaktionskonzepten ermöglicht. Indem sich der Entwickler auf die relevanten Aspekte seiner Arbeit konzentrieren kann, ohne sich von grundlegenden Implementationsproblemen aufhalten zu lassen, welche ihm durch das MedioVis-Framework abgenommen werden, soll seine Innovativität gesteigert und der Entwicklungsprozess beschleunigt werden.

Auf Seite der *Medio Vis-Applikation* ist das Ziel, eine benutzerzentrierte Anwendung zu implementieren. Um dies umzusetzen werden zum Beispiel Evaluationsstudien durchgeführt [JGK⁺05]. Eine weitere Möglichkeit an Informationen über Benutzerverhalten zu kommen, bietet sich durch den Live-Einsatz von MedioVis in der *Bibliothek der Universität Konstanz*. Mit der Integration von DROID¹¹ [Jet03], einem weiteren Projekt der *Arbeitsgruppe*

¹⁰<http://www.ub.uni-konstanz.de/>

¹¹<http://hci.uni-konstanz.de/index.php?a=research&b=projects&c=3042484&lang=de>

Human-Computer Interaction, werden während der realen Benutzung Interaktionsdaten mitgeschnitten. Mit Hilfe dieser Daten können dann Langzeitstudien zur Benutzung von MedioVis und die Auswirkung von Veränderungen in den Visualisierungen und dem Interaktionskonzept durchgeführt werden.

3.3 Anwendungsdomäne

Die Hauptanwendungsdomäne von MedioVis findet sich — wegen der hervorragenden Möglichkeit im Mediotheks-Bereich der *Bibliothek der Universität Konstanz* real eingesetzt zu werden — im Gebiet der Bibliothekskataloge. Diese bieten ausserdem eine interessante Datengrundlage, da der Bibliothekskatalog einen gut gepflegten, heterogenen Informationsraum aufspannt, mit dem verschiedene Konzepte im realen Umfeld getestet werden können. Im Falle der *Bibliothek der Universität Konstanz* handelt es sich um einen Katalog, der mit über 2 Millionen Titeln auch eine beachtliche Grösse aufweist. Durch die Einschränkung auf den Bereich der Mediothek wird die Grösse des Kataloges auf rund 60'000 Titel begrenzt, wobei innerhalb der Arbeitsgruppe immer auch die volle Datenbank verwendet wird, um die Skalierung des Systems auf grosse Datenbanken sicherzustellen. MedioVis kann dabei als alternatives Suchsystem zum konventionellen Online-Katalogsystem KOALA¹² benutzt werden. KOALA ist ein auf HTML-Formularen und Hyperlink-Strukturen basierendes Online-Suchsystem, welches auf dem Bibliotheks-Management-System LIBERO¹³ basiert.

Eine besondere Stärke von MedioVis ist, dass auf sehr einfache Art und Weise neue Datenquellen an das System angeschlossen werden können. Darum werden neben den Mediotheksdaten auch immer wieder neue Anwendungsdomänen ausprobiert. Unter anderem wurde die Internet Movie Database (IMDb)¹⁴, die Web-Suchmaschine YAHOO!¹⁵, das Internetauktionshaus eBay¹⁶, Peer-2-Peer Netzwerke aber auch kleinere Kataloge und Datensammlungen in das MedioVis-Framework integriert und dieses mit den entsprechenden Interaktionsmöglichkeiten ausgestattet.

¹²KOALA: Konstanzer Ausleih- und Anfrage-System, <http://www.ub.uni-konstanz.de/koala/>

¹³<http://www.libero.com.au/>

¹⁴<http://imdb.com/>

¹⁵<http://www.yahoo.com/>

¹⁶<http://www.ebay.com/>

3.4 Definition des Datenraumes

In dieser Arbeit soll auf die Domäne des Mediothek-Kataloges zurückgegriffen werden, welcher sich durch seine Heterogenität und die Multidimensionalität der einzelnen Datenelemente als geeignete Datenbasis auszeichnet. Die sich in der Mediothek befindlichen Datensätze lassen sich jeweils einem der Medientypen *Videokassette*, *DVD*, *CD*, *Tonträger*, *Buch*, *Zeitschrift*, *Karte*, *Mikrofilm* und *Diskette* zuordnen. Diejenigen Datensätze, für die keine dieser Kategorien treffend erscheint, bzw. bei denen der Medientyp unbekannt ist, werden im Typ *Diverses* zusammengefasst.

Ein einzelner Datensatz beinhaltet wiederum verschiedene Attribute. Dazu gehören dessen *Titel*, falls vorhanden dessen *Originaltitel*, eventuelle weitere *Untertitel*, eine Aufzählung von beteiligten *Personen* (bei Filmen sind dies zum Beispiel die Schauspieler und der Regisseur), das *Jahr der Veröffentlichung*, die *Sprache*, der *Herausgeber*, der *Veröffentlichungsort*, das dem Datensatz zugeordnete *Fachgebiet*, die *Signatur* des Datensatzes um es in der Bibliothek auch physisch auffinden zu können, das *Format* (bei DVDs zum Beispiel die Länderkodierung und ähnliches), die *Ausleihfrequenz* welche als Mass für die Beliebtheit des Datensatzes eingesetzt wird und zwei Attribute mit weiteren *Anmerkungen* und *Details* zum Datensatz. Alle diese Attribute werden Serverseitig in einem nächtlichen Durchgang, auf den aktuellen Stand gebracht, so dass jeden Morgen alle neu in den Mediotheks-Katalog aufgenommenen Datensätze dem MedioVis-Nutzer zur Suche und zur Exploration zur Verfügung stehen, bzw. alle ausgemusterten Medienträger auch aus der MedioVis-Datenbank verschwunden sind.

Durch das geschickte Abfragen von meist externen Webressourcen werden die von der Bibliothek gelieferten Daten angereichert. Beispielsweise wird über ein auf dem Bibliotheksserver laufendes Skript, für jeden Datensatz der aktuelle *Ausleihstatus* abgefragt und als weiteres Attribut genutzt. Da gerade diese Information sich schnell ändern kann, muss dies zur Runtime im MedioVis-Client geschehen.

Weitere externe Ressource die genutzt werden, sind die Webservices Amazon.com¹⁷ und die IMDb, über welche nach Buchcovern und Filmpostern gesucht wird. Dadurch ist es möglich, ein Attribut namens *Poster* in die Datenbasis mit aufzunehmen.

¹⁷<http://amazon.com/>

In die gleiche Kategorie von aus externen Quellen angereicherten Daten gehören die von der IMDb abgegriffenen Zusatzinformationen zu Filmen und Schauspielern. So können zum Beispiel Personenbiographien in die Visualisierungen von MedioVis integriert werden. Auch können weitere externen Webservices wie Google-Maps¹⁸ oder Links zu Wikipedia¹⁹ mit den bereits vorhandenen Daten vereint werden. Abbildung 10 zeigt eine solche Forschungsversion von MedioVis. In einer Visualisierung namens HyperGrid, auf die in Kapitel 4.2 genauer eingegangen wird, werden zusätzlich zu den eigentlichen Katalogdaten Filmposter von Amazon.com, Benutzerkommentare von der IMDb-Webseite, geographische Informationen von Google-Maps, ein Filmtrailer und Informationen von Wikipedia mit eingebunden.



Abbildung 10: In den MedioVis-Client können neben den eigentlichen Katalogdaten auch Daten von externen Quellen, wie Amazon, Google-Maps oder der IMDb integriert werden (hier am Beispiel der HyperGrid).

Um den Benutzer im Mediotheks-Kontext noch weiter unterstützen zu können, wird ihm als weiteres Attribut zu jedem Datenobjekt ein Lageplan der Mediothek angeboten. Darauf

¹⁸<http://maps.google.com/>

¹⁹<http://wikipedia.org/>

ist rot das Regal markiert, in dem das physische Exemplar zu finden ist (siehe Abbildung 11).

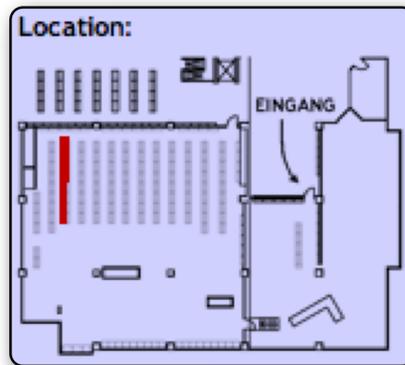


Abbildung 11: Das Attribut Location bzw. Lageplan.

Es ist auch möglich ganze Filmtrailer, PDF-Dokumente, Musik-Dateien und ähnliches in die Attributmenge aufzunehmen (im Prinzip bis hin zum eigentlichen digitalisierten Medienträger selbst). Im Rahmen der öffentlich genutzten MedioVis-Version muss darauf aber wegen Copyright-Gründen verzichtet werden.

3.5 Definition des Use-Cases für diese Arbeit

In dieser Arbeit soll von einem typischen Mediotheks-Szenario ausgegangen werden. Ein genaues Anwendungsszenario wird in den Kapiteln 4.1 und 5.4 vorgestellt. Es wird auf den oben vorgestellten Mediotheksdaten gearbeitet, wobei durchaus externe Informationsquellen eingebunden werden sollen. Das Ziel soll sein, ein visuelles Suchsystem vorzustellen, welches das Auffinden von Datensätzen der Mediothek und die Analyse des gesamten Datenraums ermöglicht. Das Benutzerprofil wird entsprechend des Einsatzgebietes, also der Mediothek bzw. *Bibliothek der Universität Konstanz*, definiert.

4 Klassische Benutzung von MedioVis

Im Folgenden soll mittels einem Anwendungsszenario dargestellt werden, wie eine typische Benutzung der aktuellen MedioVis-Version (ohne die neue HyperScatter-Visualisierung) aussehen würde. Dabei wird von einer Konfiguration des MedioVis-Clients ausgegangen, die zwei bereits vorhandene Visualisierungen beinhaltet, die HyperGrid [JGK⁺05] und den ScatterPlot (siehe Abbildung 12). Diese beiden Visualisierungen werden danach in den Abschnitten 4.2 und 4.3 genau beschrieben.

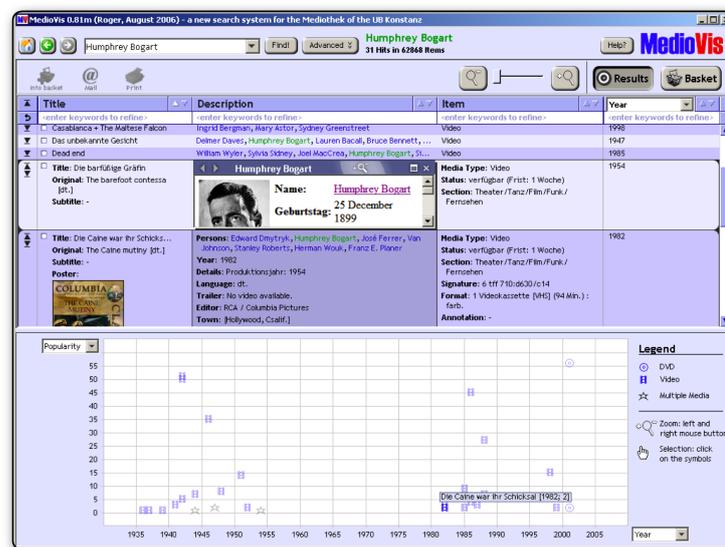


Abbildung 12: MedioVis-Client mit den Visualisierungen HyperGrid (im oberen Bereich) und ScatterPlot (im unteren Bereich).

4.1 Anwendungsszenario Teil I (HyperGrid und ScatterPlot)

Kathrin Schanzer²⁰ studiert an der Universität Konstanz im vierten Semester Biologie. Sie kennt sich mit Computern aus, da sie als Studentin beim Lernen und Recherchieren öfters mit dem Internet als Informationsquelle in Berührung gekommen ist.

Neben dem Biologie-Studium hat sie sich privat schon immer für Film- und Medienwissenschaften interessiert und würde sich gerne einmal als Regisseurin ihres eigenen Filmes

²⁰Fiktive aber charakteristische MedioVis-Benutzerin

versuchen. Aus diesem Grund hat sie sich entschieden die Vorlesung „Wie funktioniert ein Drehbuch?“ vom Fachbereich Literaturwissenschaften zu besuchen. Dabei muss sie als Leistungsnachweis ein Referat vorbereiten, in welchem sie die Zusammenhänge zwischen verschiedenen Drehbucharten und den resultierenden Filmen aufzeigen soll. In diesem Referat will sie natürlich auch Ausschnitte aus den von ihr analysierten Filmen zeigen. Bereits früh im Semester entscheidet sie sich, schon einmal nach geeigneten Filmen in der Mediothek zu suchen, um sich diese in Ruhe anschauen zu können.

In der Mediothek setzt sie sich deshalb an einen öffentlich zugänglichen PC-Arbeitsplatz auf dem MedioVis installiert ist, welches sie bereits früher schon einmal genutzt hat. Sie startet die Anwendung und findet sich schnell wieder in MedioVis zurecht, weiss, wie sie mittels Zooming (durch Drücken der linken und der rechten Maustaste auf einer Tabellenzeile) in der HyperGrid an die Detailinformationen zu den gefundenen Datensätzen kommt. Auch versteht sie das Prinzip des ScatterPlots.

Als erstes ist Kathrin am Film *Psycho* von *Alfred Hitchcock* interessiert. Sie tippt den Titel in das Suchfeld am oberen Rand des Startbildschirmes ein und schickt die Suche ab. Nach kurzer Zeit wird ihr der Resultatbildschirm angezeigt (siehe Abbildung 13, oben links). Die Anwendungsoberfläche ist in zwei grosse Bereiche eingeteilt. Direkt unter dem Suchfeld befindet sich die HyperGrid. Darunter werden die Suchtreffer im ScatterPlot ein zweites mal, parallel zur HyperGrid, visualisiert. Am oberen Rand, rechts des Suchfeldes befindet sich die Statusanzeige, welche jetzt darauf hinweist, dass zur getätigten Suchanfrage 9 Treffer aus 62'868 Datensätze der Mediothek gefunden wurden.

Kathrin entdeckt in der HyperGrid sofort die beiden Datensätze die den Titel *Psycho* tragen und die offensichtlich etwas mit *Alfred Hitchcock* zu tun haben, da dieser Name in der zweiten Spalte zu sehen ist (siehe Abbildung 13, oben links). Auch kann sie sofort erkennen, dass es sich bei einem der beiden Datensätze um eine Videokassette und beim anderen um eine DVD handelt. Jetzt möchte sie die beiden Suchtreffe genauer untersuchen um weitere Unterschiede festzustellen. Um an mehr Detailinformationen zu kommen, zoomt sie, durch Drücken der linken Maustaste auf die entsprechende Tabellenzeile, in die Datensätze hinein. Leider ist es aufgrund des beschränkten Platzes auf dem Computerbildschirm nicht möglich, beide Datensätze gleichzeitig im vollständig gezoomten Zustand anzuzeigen und somit vergleichen zu können. Kathrin bemerkt jedoch, dass sie den ScatterPlot im unteren Bereich des Bildschirms verkleinern kann, indem sie den Begrenzungsbalken zwischen den beiden Visualisierungen HyperGrid und ScatterPlot



Abbildung 13: Klassisches MedioVis-Anwendungsszenario (mit den Visualisierungen HyperGrid und ScatterPlot).

zum unteren Bildschirmrand hin verschiebt (siehe Abbildung 13, oben rechts). Durch den Platzgewinn zugunsten der HyperGrid können nun die Detailinformationen von beiden interessanten Datensätzen gleichzeitig angezeigt werden. Schnell hat sie sich nun für die DVD mit englischem Originalton entschieden und holt sich diese anhand des Lageplans, welcher nach einer erneuten Zoom-Interaktion ebenfalls angezeigt wird, aus dem Regal.

Als sich Kathrin wieder an den Computer-Arbeitsplatz setzt, kommt ihr die Idee, dass auch noch andere Hitchcock-Filme von Interesse sein könnten. Insbesondere wären dies die eher jüngeren Filme von *Alfred Hitchcock*, welche in englischem Originalton zur Verfügung stehen. Wenn möglich sollten sie als DVD vorhanden sein, damit sie bei ihrem Referat nicht einen DVD-Player und einen VHS-Player organisieren muss. Sie startet also eine neue Suche nach *Hitchcock* und bekommt das Suchresultat wiederum nach einer kurzen Wartezeit präsentiert (siehe Abbildung 13, Mitte).

Kathrin erinnert sich an den ScatterPlot und möchte diesen jetzt gerne nutzen, um die relevanten Datensätze schnell in der Treffermenge ausfindig zu machen. Dazu schiebt sie den Begrenzungsbalken zwischen HyperGrid und ScatterPlot wieder nach oben (siehe Abbildung 13, unten links). Nachdem sie im ScatterPlot die *Sprache* als Attribut auf die Y-Achse gelegt hat, indem sie den entsprechenden Eintrag aus der Dropdown-Box neben der Achse ausgewählt hat, verschiebt sie den Begrenzungsbalken zwischen den beiden Visualisierungen so, dass alle vorhandenen Ausprägungen von Sprach-Kombinationen angezeigt werden können. Nun hat sie schnell die für sie relevanten Datenpunkte ausgemacht. Diese finden sich im rechten Bereich des ScatterPlots, wo sich die neueren Werke von und über Alfred Hitchcock befinden, auf den beiden Y-Achsenabschnitten *engl.* und *engl.+dt.*, welche englischem Originalton entsprechen. Auch sieht Kathrin sofort, dass sich unter den in Frage kommenden Datensätzen keine DVDs befinden, sondern nur Videokassetten. Trotzdem möchte sie jetzt zu besagten Videos die Detailinformationen anschauen, um herauszufinden, ob sich der technische Mehraufwand für das Referat vielleicht doch lohnen würde.

Sie entdeckt nun, dass durch Klicken auf einen Datenpunkt im ScatterPlot ein Pie-Menu erscheint, mit dem sie die HyperGrid „fernsteuern“ kann. Es ist damit möglich, in der HyperGrid in die Details von Datenobjekten hinein zu zoomen. Mittels des Pie-Menus versucht sie in die Detailinformationen der interessanten Datensätze hinein zu zoomen, muss aber feststellen, dass in der HyperGrid nicht genügend Platz zur Verfügung steht, um alle für sie relevanten Attribute auf einmal darstellen zu können.

Wohl oder Übel entscheidet sie sich, den ScatterPlot wieder zu verkleinern, um wieder möglichst viel Platz für die HyperGrid zur Verfügung zu haben. Anschliessend geht sie die Tabelle Zeile für Zeile durch, um die relevanten Datensätze ausfindig zu machen (siehe Abbildung 13, unten rechts). Immerhin kann sie dabei von den Spaltenfiltern der HyperGrid Gebrauch machen. Dazu wählt sie in der Dropdown-Liste in der vierten Spalte *Sprache* aus und kann nun alle Datensätze ausblenden, die nicht auf das Filterkriterium *engl.* zutreffen. Leider werden so immer noch diejenigen Filme mit angezeigt, die nur englische Untertitel aufweisen, nicht aber im englischen Originalton zur Verfügung stehen. Das manuelle Aussortieren dieser Treffer und das Analysieren des Erscheinungsjahrs der verschiedenen Filme nimmt also wesentlich mehr Zeit in Anspruch, als im ScatterPlot, in dem aber die Detailinformationen nicht direkt zugänglich sind.

4.2 Die HyperGrid

In diesem Abschnitt soll die im beschriebenen Anwendungsszenario präsentierte HyperGrid und im folgenden Abschnitt der ScatterPlot genauer beschrieben werden, da viele der diesen beiden Visualisierungen zu Grunde liegenden Konzepte in den im Hauptteil dieser Arbeit beschriebenen HyperScatter eingehen. Dabei stellt das oben beschriebene Anwendungsszenario das klassische MedioVis dar, also die MedioVis-Konfiguration mit HyperGrid und ScatterPlot. Der später beschriebene HyperScatter und dessen Integration in das vorhandene visuelle Suchsystem entsprechen einem alternativen Vorschlag eines Interaktions- und Visualisierungskonzepts für MedioVis.

Die HyperGrid hat ihren Namen von zwei ihr zu Grunde liegenden Konzepten. Zum Einen ist sie in der Lage, hypermediale Inhalte darzustellen. Im Beispiel vom Mediothekskontext können zum Beispiel zu einem Film nebst textuellen Metadaten auch Filmposter und Videotrailer in die HyperGrid integriert werden. Zusätzlich ist es möglich, zu den präsentierten Metadaten eines Films weitere externe Datenquellen anzubinden, wie beispielsweise Schauspielerbiographien. Dies führt dazu, dass der Aspekt des Browsens in dem dargestellten Informationsraum eine wesentliche Rolle einnehmen kann. Zum Anderen weist die HyperGrid eine zweidimensionale, tabellenartige Struktur auf. Obwohl sie auf den ersten Blick sehr stark an eine gewöhnliche Tabelle erinnert, ist diese Struktur aber nicht steif wie bei einer Tabelle, sondern die Grössen der einzelnen Zellen sind durch Benutzerinteraktionen variabel wählbar, was daher eher an eine flexible, netzartige Struktur erinnert.

Im Gegensatz zu den eher schwach strukturierten Listendarstellungen, welche eine weit verbreitete Art der Trefferrepräsentation in Suchsystemen wie Katalog- oder Websuchen ausmachen, stellt eine tabellenartige Suchtrefferdarstellung durch ihre starke Strukturierung der angezeigten Informationen eine der Listendarstellung oft überlegene Repräsentation von Suchresultaten dar [Ger04]. Hierbei werden die einzelnen Attribute eines Suchtreffers jeweils in eine eigene Spalte der Tabelle gesetzt. Mechanismen wie das Sortieren oder das nachträgliche Filtern der Suchtreffermenge anhand einzelner Tabellenspalten, bzw. anhand der in diesen Spalten enthaltenen Attribute, ist bei Listendarstellungen, wie sie zum Beispiel von den meisten Web-Suchmaschinen verwendet werden, nicht möglich.

Probleme treten allerdings auf, wenn die Anzahl der darzustellenden Attribute, respektive die Anzahl der Tabellenspalten, so gross ist, dass nicht die gesamte Tabelle auf einmal dargestellt werden kann. In diesem Fall muss der Benutzer horizontal durch die Tabelle navigieren oder er muss die Anzahl der momentan dargestellten Tabellenspalten, entsprechend seinem momentanen Informationsbedürfnis, weiter einschränken.

4.2.1 Aspect of Interest

In der HyperGrid wird das im letzten Absatz beschriebene Problem so gelöst, dass mehrere Attribute in einer Tabellenspalte zusammengefasst werden. Dabei werden einzelne Datenobjekte, wie bei gewöhnlichen Tabellen, weiterhin durch eine Tabellenzeile gekapselt. Die HyperGrid verwendet aber nur eine beschränkten Anzahl an Tabellenspalten, in welchen semantisch ähnliche Informationen bzw. Attribute gruppiert werden. Jede Spalte repräsentiert also „einen“ *Aspect of Interest* [JGK⁺05] (siehe Abbildung 14).

Im Beispiel Handelt es sich um die drei Aspects of Interest: *Title*, *Description* und *Item*. In der Spalte *Title* werden diejenigen Attribute zusammengefasst, welche den Inhalt des repräsentierten Medienobjekts charakterisieren, also der Titel, der Originaltitel, der Untertitel und das Filmposter. In der Spalte *Description* werden Attribute in eine Gruppe gefasst, welche die technische Seite des Medienobjektes beschreiben. Das sind zum Beispiel Attribute wie die beteiligten Personen (Schauspieler, Regisseure etc.), das Erscheinungsjahr, die Sprache des Medienobjektes und der Produktionsort. In der Spalte *Item* werden schliesslich die Attribute zusammengefasst, welche Informationen über das physische Exemplar, welches in der Mediothek im Regal steht, enthalten. Dies sind Attribute wie der Medientyp (Videokassette, DVD etc.), die Signatur des Exemplars oder auch der Ausleih-

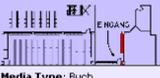
Title	Description	Item	Year
<input type="checkbox"/> Title: Andromeda - Tödl... Original: The Andromeda ... Subtitle: - Poster: 	Persons: Robert Wise, Arthur Hill, David Wayne, James Olson, Michael Crichton, Nelson Gidding, Richard H. Kline, ... Year: 2002 Details: Spielfilm, Literaturverfilmung, USA 1970. - Zweikanalton: dt./engl. Language: dt.+engl. Trailer: No video available. Editor: Bayerisches Fernsehen Town: [München] Institutions: -	Media Type: Video Status: verfügbar (Frist: 1 Woche) Section: Theater /Tanz/Film/Funk /Fernsehen Signature: 6 tff 710:w813/a62 Format: 1 Videokassette (VHS, 125 Min.) : farb. Annotation: Fernsehmitschnitt Location: 	2002
<input type="checkbox"/> Title: Basic econometrics Original: - Subtitle: - Poster: 	Persons: Damodar N. Gujarati Year: 2003 Details: CD-ROM enthält zahlr. Beispiele u. Tabellen Language: engl. Trailer: No video available. Editor: McGraw-Hill Town: Boston [u.a.] Institutions: -	Media Type: Buch Status: mehrere Exemplare Section: Wirtschaftswissenschaften/Met... Signature: 6 wrrh 300:n/g94a(4)-cdrom... Format: XXIX, 1002 S. : graph. Darst. Annotation: 720 ddsu/sfu,spz,stb ; 720 ff. (DDC); GBV/LOC; 5090 d... Location: 	2003
<input type="checkbox"/> Title: Basic vocabulary Original: - Subtitle: - Poster: -	Persons: Eugene Ehrlich, Daniel Murphy Year: 1975 Details: - Language: - Trailer: - Editor: - Town: - Institutions: -	Media Type: Buch Status: - Section: - Signature: - Format: - Annotation: - Location: -	1975

Abbildung 14: Ausschnitt aus der HyperGrid zur Veranschaulichung des Konzepts von Aspect of Interest.

status, welcher anzeigt, ob das Exemplar im Moment verfügbar ist, oder ob es gerade von einem anderen Mediotheksbenutzer ausgeliehen ist.

Um weiterhin die Möglichkeit zu haben, nach jedem einzelnen Attribut sortieren und filtern zu können, gibt es in der HyperGrid noch eine weitere Spalte, die so genannte *User Adjustable Column* [JGK⁺05]. Hier kann der Benutzer jedes beliebige Attribut durch Auswählen von einer Dropdown-Box in die letzte Spalte legen. In Abbildung 14 ist zum Beispiel das Erscheinungsjahr ausgewählt. So bietet sich ihm die Möglichkeit, die Tabelle nach jedem beliebigen Attribut zu sortieren, beziehungsweise den Spaltenfilter, welcher später noch genauer erklärt wird, zu nutzen um gezielt nach einem speziellen Attribut zu filtern.

4.2.2 Degree of Interest und Level of Detail

Nach dem ein Benutzer eine Suchanfrage über den MedioVis-Client abgeschickt hat, werden ihm die Suchresultate in einer möglichst kompakten Form dargestellt, damit er eine gute Übersicht über die Resultatmenge bekommen kann. In der HyperGrid ist es dazu möglich die einzelnen Tabellenspalten und Zellen so zusammen zu klappen, dass ein einzelnes Datenobjekt nur noch eine einzige Textzeile in Anspruch nimmt (siehe Abbildung

15, oben links). Dies entspricht zugleich der Standarddarstellung, welche unmittelbar nach einer Suche zur Darstellung der Suchresultate aufgebaut wird. Der Trade-Off davon ist, dass so nicht mehr alle Detailinformationen zu den einzelnen Treffern dargestellt werden können.



Abbildung 15: Von oben links nach unten rechts: Der Level of Detail des Suchtreffers „Casablanca“ wird durch eine Zoom-Interaktion des Benutzers erhöht, was sich in dem Anwachsen der entsprechenden HyperGrid-Zeile manifestiert.

Wenn nun ein Benutzer mehr Informationen zu einem bestimmten Datenobjekt benötigt, also einen erhöhten Degree of Interest [RC94] zu diesem Datenobjekt äussern will, kann er durch anhaltendes Drücken der linken Maustaste in die entsprechende HyperGrid-Zeile den Level of Detail kontinuierlich erhöhen. Die entsprechende Zeile wird grösser gezoomt (siehe Abbildung 15), und auf dem so entstehenden Platz werden nun mehr Information zu diesem Datenobjekt angezeigt. In der HyperGrid ist also der Degree of Interest direkt dem Level of Detail zugeordnet. Je länger die Zoominteraktion andauert (wobei diese auch in mehreren Etappen ausgeführt werden kann), desto mehr Attribute des in dieser Zeile repräsentierten Datenobjekts können in der HyperGrid dargestellt werden. Zu beachten gilt jedoch, dass durch den erhöhten Platzbedarf von einzelnen Datenobjekten, deren Level of Detail erhöht

wurde, andere Datenobjekte aus dem Anzeigebereich des Bildschirms verschwinden. Diese Datenobjekte sind dann nur noch durch vertikales Scrollen der Treffermenge erreichbar.

Analog zum Erhöhen des Level of Detail, wird durch einen Rechts-Klick auf eine aufgezoomte HyperGrid-Zeile diese wiederum verkleinert, so dass immer weniger Informationen zu diesem einzelnen Datenobjekt angezeigt werden. Dabei wird der so entstandene Platz in der HyperGrid wieder für das Darstellen von vorher am unteren und oberen Bildschirmrand verschwundenen Datenobjekte genutzt.

Hinein- und hinausgezoomt wird in der HyperGrid aber nicht nur auf Zeilenbasis. Der Benutzer steuert zugleich auch die Breite der einzelnen HyperGrid-Zellen durch die oben beschriebene Mausinteraktion. Das heisst, dass durch das Links-Klicken in eine HyperGrid-Zelle einerseits der Level of Detail des zugehörigen Datenobjekts erhöht wird (also die vertikale Dimension der Zeile), andererseits aber zugleich auch der Platz der Spalte, und somit des zugehörigen Aspects of Interest (also die Breite der Spalte) vergrössert wird. Dies geschieht natürlich auf Kosten der restlichen Spalten, welche dabei schmaler werden. Der Benutzer drückt also durch eine einzige Mausinteraktion einen höheren Degree of Interest für einen speziellen Aspect of Interest eines einzelnen Datenobjekts aus.

4.2.3 HyperGrid-Browser

Viele Informationsräume beinhalten Daten, die in Beziehung zu einander stehen. Im Rahmen des in dieser Arbeit betrachteten Beispiels der Mediodaten sind solche Beziehungen zum Beispiel über die an Filmen beteiligten Personen sichtbar. Der Regisseur eines Filmes hat meistens noch bei anderen Filmen Regie geführt. Das selbe gilt auch für Schauspieler. Obwohl diese Personen nicht den primären Datenraum des Anwendungsszenarios darstellen, können sie für den Benutzer von grossem Interesse sein. Einerseits kann sich jemand für die zusätzlichen Informationen der einzelnen beteiligten Personen eines Filmes interessieren, also für deren Biographien etc., andererseits kann über den entstehenden Browsing-Ansatz eine Inspiration des Benutzers erfolgen, durch die weitere Filme gefunden werden, welche durch die ursprüngliche Suche nicht abgedeckt wurden.

In der HyperGrid ist es deshalb durch die Verwendung von Hypertext möglich, solche Beziehungen und Netze in der Visualisierung abzubilden. Damit ein Benutzer den Kontext zur ursprünglichen Trefferdarstellung und zu den einzelnen Datenobjekten nicht verliert, wird

nicht wie bei herkömmlichen Informationssystemen der Ansatz verfolgt, die zusätzlichen Informationen in neuen Fenstern darzustellen oder den Inhalt des aktuellen Anwendungsfensters durch den neuen Inhalt auszutauschen. Die verlinkten Informationen, wie zum Beispiel Schauspielerbiographien, werden in einem Browserfenster dargestellt, das in der entsprechenden HyperGrid-Zelle residiert (siehe Abbildung 16).



Abbildung 16: Von oben links nach unten rechts: HyperGrid ohne Browser — HyperGrid mit Schauspielerbiographie in einem HyperGrid-Browser — maximierter HyperGrid-Browser — mehrere HyperGrid-Browser zum Teil mit externen Webseiten.

Da in einer HyperGrid-Zelle nur ein sehr begrenzter Raum zur Darstellung des Browserfensters zur Verfügung steht, wird dem Benutzer die Möglichkeit geboten das Browserfenster auf den ganzen Bildschirmplatz der HyperGrid zu vergrößern (siehe Abbildung 16, unten links). Dabei überlagert das Browserfenster temporär die eigentliche HyperGrid. Dieses Maximieren des HyperGrid-Browsers ist jedoch nur nötig, wenn derart viele Informationen angezeigt werden, dass die Zellengröße in der HyperGrid keinen ausreichend grossen Rahmen für das Browserfenster mehr bieten kann. Durch einen einzigen Mausklick auf

das X in der Browserfensterleiste wird der Browser mit den weiterführenden Informationen wieder auf HyperGrid-Zellengrösse verkleinert und die normale Trefferdarstellung hergestellt. Um dem Benutzer das Nachvollziehen dieser Änderungen der Informationsdarstellung zu erleichtern, wird das Grösser- und Kleiner-Zoomen des HyperGrid-Browsers analog zum Zoomen in der normalen HyperGrid-Zelle animiert.

Im Beispiel von Abbildung 16 hat der Benutzer beim Film *Casablanca* auf den als Hyperlink gekennzeichneten Namen der Schauspielerin *Ingrid Bergman* geklickt (Abbildung 16, oben links). Dadurch ist in der entsprechenden HyperGrid-Zelle ein Browserfenster mit Zusatzinformationen zur Schauspielerin aufgegangen (Abbildung 16, oben rechts). Diese Informationen haben zwar nicht direkt etwas mit dem zentralen Datenobjekt, dem Film *Casablanca* zu tun, jedoch können diese weiterführenden Informationen für den Benutzer durchaus von Interesse sein. Im Beispiel sind, wie in Abbildung 16 unten links zu erkennen ist, auch alle Filme von *Ingrid Bergman* aufgeführt. Dabei sind diese wieder als Hyperlinks gekennzeichnet. Über diese Links wird dem Benutzer ermöglicht, direkt eine neue Suche auf den Mediodaten auszuführen. So würde zum Beispiel bei einem Mausklick auf den zweiten Film in der Liste nach *A Matter of Time* gesucht werden. Die Suchtreffer würden dann wieder in der HyperGrid dargestellt.

Was genau im Browserfenster angezeigt wird, kann, je nach Use-Case in dem die HyperGrid genutzt wird und natürlich nach dem Element das verlinkt wurde, frei definiert werden. Dabei können auch beliebige externe Web-Ressourcen wie zum Beispiel Google-Maps, Wikipedia oder die IMDb angezeigt werden. Im Beispiel von Abbildung 16 werden nebst diversen Links zu Suchdiensten und Informationsplattformen, wie unten gut zu erkennen ist, auch Google-Maps dazu verwendet, um den Geburtsort von Personen zu visualisieren. Ist zu Beginn von *Ingrid Bergman* nur die textuelle Geburtsortangabe *Stockholm, Sweden* zu sehen, so wird durch einen Mausklick auf den Hyperlink ein Satellitenbild von Skandinavien in den Browser geladen, in dem die Stadt Stockholm durch eine rote Markierung hervorgehoben ist (unten rechts).

Bei Bedarf besteht also die Möglichkeit, beliebig tief in den Informationsraum einzutreten. Es wird dem Benutzer durch den HyperGrid-Browser ein Einstiegspunkt in den praktisch unendlich grossen Informationsraum des World Wide Webs geboten. Dabei ist aber jedes geöffnete Browserfenster ganz klar in einer Zelle der HyperGrid-Tabelle verankert. So verliert der Benutzer selbst dann nicht den Kontext zu seiner eigentlichen Suche in den Mediodaten, beziehungsweise zum eigentlichen Datenobjekt, wenn er bereits

tief in das Informationsangebot eingetaucht ist. Das Angebot an Informationen, welche über die HyperGrid-Browser erreichbar sind, kann aber auch ganz einfach eingeschränkt werden, indem von der ersten angezeigten Seite an nur kontrollierte Hyperlinks angeboten werden, welche selber wieder nur gewünschte oder sogar keine weiteren Hyperlinks enthalten.

4.2.4 Spaltenfilter und Sortierung

Ein weiteres sehr wichtiges Feature der HyperGrid ist der Spaltenfilter (siehe Abbildung 17). Im Beispiel wird der Filter der User Adjustable Column verwendet, um dynamisch alle Datenobjekte aus der Treffermenge weg zu filtern, deren Titel nicht auf das Filterkriterium „cas“ zutreffen. Durch das Eintippen von nur wenigen Zeichen, kann so eine grosse Treffermenge auf nur noch sehr wenige relevante Datenobjekt reduziert werden.



Abbildung 17: HyperGrid-Spaltenfilter: Aus der Treffermenge werden sukzessive alle Datenobjekte ausgefiltert, deren Titel nicht auf das Filterkriterium „c“, „ca“, „cas“ passen.

Spaltenfilter sind auch in den restlichen Spalten der HyperGrid verfügbar. Bei diesen wird dann sinngemäss in allen Attributen die in der Spalte zusammengefasst werden nach dem Filterkriterium gesucht. Hier wird nochmals die Wichtigkeit der User Adjustable Column klar, mit der es erst möglich wird, nur nach einem ganz bestimmten Attribut, wie Titel oder Erscheinungsjahr, zu filtern.

Neben dem Spaltenfilter ist in den Tabellenkopf der HyperGrid auch ein Button eingebaut, mit dem der Benutzer in der Lage ist, die Treffer Anhand des Inhaltes der Spalte zu sortieren. Dies geschieht so, wie von Programmen wie Dateimanagern oder Tabellenkalkulationen bekannt. In Abbildung 17 ist die Treffermenge zum Beispiel alphabetisch, aufsteigend nach dem Titel sortiert. Auch hier kann die User Adjustable Column da-

zu verwendet werden, nach einem bestimmten Attribut zu sortieren. Bei den restlichen HyperGrid-Spalten wird jeweils nach dem obersten Attribut in der Zelle sortiert.

Die beiden Möglichkeiten, in der HyperGrid die Treffermenge nachträglich nach beliebigen Attributen zu sortieren und in beliebigen Attributen im Sinne von Dynamic Queries [AWS92] zu filtern, ermöglicht dem Benutzer, ohne das erneute Formulieren von Suchanfragen, die Treffermenge weiter einzuschränken, um aus allen Suchtreffern die wirklich relevanten Datenobjekte zu extrahieren.

4.2.5 Extraktion von Detailinformationen aus der HyperGrid

MedioVis soll den Benutzer natürlich auch nach der eigentlichen Suche und Analyse im Katalog bei seiner Arbeit unterstützen. Dazu ist es nötig, dass die gefundenen Informationen zu den für den Benutzer relevanten Datenobjekten auch ausserhalb von MedioVis genutzt und weiterverarbeitet werden können. Dazu stellt MedioVis verschiedene Tools zur Verfügung.

Als erstes ist es in der HyperGrid möglich, jedes angezeigte Datenobjekt zu selektieren. Dies geschieht über eine Checkbox, die in jeder HyperGrid-Zeile auf der linken Seite vorhanden ist (siehe Abbildung 18). In der Abbildung sind zum Beispiel die beiden DVDs *2001: Odyssee im Weltraum* und *Eyes wide shut* selektiert. In der Symbolleiste von MedioVis können nun verschiedene Funktionen auf die selektierten Datenobjekte angewendet werden.

Mit einem Klick auf das Symbol *Into basket* können die Datenobjekte in einen speziellen Container kopiert werden. In Anlehnung an viele Online-Shops wie Amazon.com etc., wird dieser Container Warenkorb (oder englisch Basket) genannt, auch wenn es dabei, zumindest im Kontext des Mediothekkatalogs, nicht um das käufliche Erwerben der ausgewählten Objekte geht. Durch Betätigung der beiden Buttons *Results* und *Basket* kann zwischen der Trefferansicht der aktuellen Suche und der Ansicht aller Datenobjekte, die in den Warenkorb gelegt wurden, umgeschaltet werden. Dieser wird wiederum als HyperGrid dargestellt und bietet genau die gleichen Interaktionsmöglichkeiten wie die HyperGrid der Trefferdarstellung. Der Warenkorb soll dem Benutzer die Möglichkeit bieten, sich verschiedene interessante Datenobjekte, über mehrere Suchen hinweg zusammen zu stellen und diese abzuspeichern. Es können also Datenobjekte ähnlich wie Bookmarks im Warenkorb



Abbildung 18: HyperGrid mit zwei selektierten Datenobjekten und den Funktionen zum Hinzufügen dieser Objekte zum Warenkorb, dem Verschicken der Detailinformationen per E-Mail oder der Möglichkeit des Ausdrucksens dieser Informationen.

abgelegt werden. Das heisst, dass der Warenkorb, im Gegensatz zur Trefferdarstellung, bei einer neuen Suche nicht gelöscht wird, um dann mit den neuen Suchtreffern neu gefüllt zu werden, sondern die Datenobjekte, die vom Benutzer in den Warenkorb gelegt wurden, bleiben auch über mehrere Suchen dort gespeichert.

Durch Klicken auf das Symbol *Mail* wird aus den Detailinformationen der momentan selektierten Datenobjekte ein HTML-File generiert, welches die Datensätze in einer Art Liste neu formatiert. Nachdem der Anwender eine E-Mail-Adresse angegeben hat, wird diese Liste an die Adresse verschickt. So ist es einem Benutzer möglich, sich selbst oder auch anderen Personen, die Kollektion von Datenobjekt zuzusenden um diese auch ausserhalb von MedioVis weiter zu verwenden. Ganz Ähnliches geschieht beim Klicken auf *Print*. Hier kann der Benutzer aber die Liste über einen Druck-Dialog zum nächsten Drucker schicken um die Zusammenstellung von Datenobjekt in Papierform zu erhalten, was sich im Mediotheks-Kontext gut als Checkliste bei dem, der Suche nachfolgenden, physischen Auffinden der Exemplare in den Regalen der Mediothek auszahlen kann.

4.3 Der ScatterPlot

Die zweite Visualisierung, die im Anwendungsszenario am Anfang dieses Kapitels angesprochen wurde, ist der ScatterPlot. In dieser Visualisierung werden alle Suchtreffer als kleine Symbole auf die Fläche der Visualisierung verteilt. Dabei werden sie anhand von zwei Achsen, auf die jeweils ein einzelnes Attribut gemapt ist, im Sinne eines kartesischen Koordinatensystems angeordnet (siehe Abbildung 19).

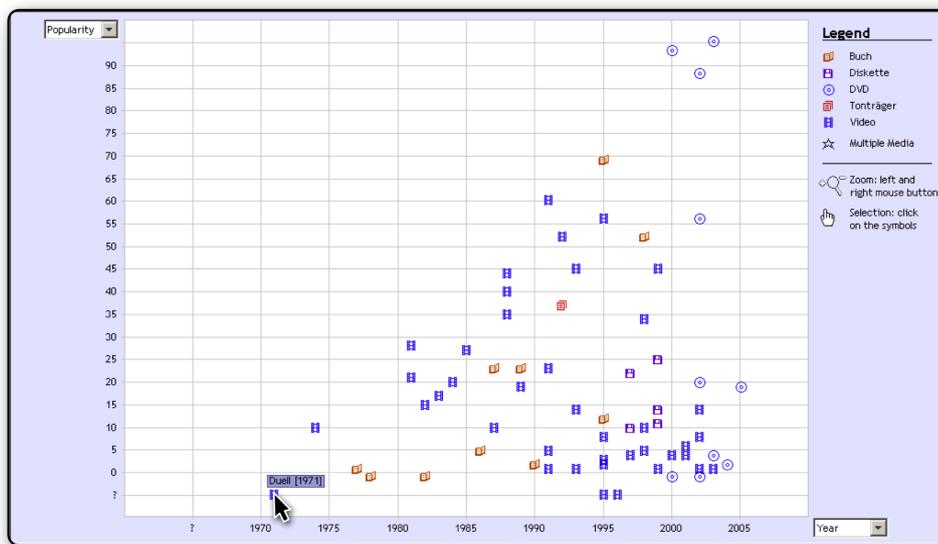


Abbildung 19: Der ScatterPlot im klassischen MedioVis-Client.

Eine Schwäche der HyperGrid ist, dass es nicht möglich ist alle Suchtreffer einer Suche auf einen Blick darzustellen, sofern diese eine gewisse Menge an Datenobjekten überschreitet. Schnell ist der Bildschirmplatz gefüllt und der Benutzer muss vertikal durch die Treffermenge scrollen, wobei er niemals alle Datenobjekte gleichzeitig im Blick haben kann. Das ist im ScatterPlot anders: Hier werden alle Datenobjekte auf einmal dargestellt. Eine Scatterplot-Visualisierung eignet sich aus diesem Grund als Übersichtsdarstellung von Suchtreffern. Eine Einschränkung die bei Scatterplots gemacht werden muss, ist dass wenn viele Datenpunkte auf einmal dargestellt werden, es zu Overplotting, also zu gegenseitiger Überdeckung einzelner Datenpunkte, kommen kann.

Im ScatterPlot von MedioVis wird neben den zwei Dimensionen (Attributen), die durch die beiden Achsen der Visualisierung abgedeckt werden eine dritte Dimension (der Medientyp) durch die Art der Repräsentation eines Datenobjekts auf dem ScatterPlot abgedeckt. Da

es sich im Fall unseres Beispiels um den Mediothekskatalog handelt, werden die einzelnen Datenobjekt nicht einfach nur als Punkte, sondern direkt als Film-, Buch-, DVD-, CD-Symbole und Ähnliches im ScatterPlot dargestellt (siehe Abbildung 19).

Ein weiterer Vorteil von Scatterplot-Visualisierungen gegenüber listen- und tabellenbasierten Darstellungen ist, dass mit ihnen Datenpunkte auf einen Blick in Relation zu mehreren Attributen gebracht werden können [Tuf83]. In Abbildung 19 ist zum Beispiel die Popularität der Suchtreffer auf die linke Achse gelegt, und am unteren Rand der Visualisierung ist eine Zeitachse mit dem Erscheinungsjahr der Datenobjekt abgetragen. Schnell können nun die populärsten Exemplare im oberen Bereich des ScatterPlots ausgemacht werden. Die neueren Exemplare befinden sich im rechten Teil der Visualisierung, die älteren eher im linken Teil. Und als Konsequenz davon befinden sich die neuen und beliebten Exemplare, die zur Treffermenge der abgeschickten Suche gehören, entsprechend in der oberen, rechten Ecke des ScatterPlots. Ausserdem lässt sich im Beispiel von Abbildung 19 erkennen, dass sich an besagter Stelle nur drei DVDs befinden. Oder anders formuliert: Auf die Suchanfrage, zu welcher die Treffermenge gehört, die in Abbildung 19 gezeigt wird, gibt es drei neue, beliebte Treffer, wobei es sich bei allen dreien um DVDs handelt. Das alles kann auf einen Blick festgestellt werden.

Ein weiteres Feature des ScatterPlots in MedioVis ist die variable Belegung der Achsen. Beide lassen sich mit beliebigen vordefinierten Attributen durch Auswählen aus einer Dropdown-Liste belegen (siehe Abbildung 19). Interessiert einen Benutzer zum Beispiel statt der Popularität der Suchtreffer, in Abhängigkeit des Erscheinungsjahres, deren Sprache, wählt er aus der Dropdown-Liste für die Belegung der Y-Achse (oben links) anstatt *Popularity* das Attribut *Language* aus. Die Suchtreffer arrangieren sich unmittelbar um, und schon können zum Beispiel alle Datenobjekte erkannt werden, die in deutscher Sprache vorliegen und zwischen 1990 und 1995 erschienen sind.

Allerdings ist es im ScatterPlot von MedioVis nicht möglich, direkt an die Detailinformationen der einzelnen Datenobjekte zu gelangen. Es können jeweils nur die zwei Attribute mit denen die beiden Achsen belegt sind und der Medientyp der Datenobjekte direkt aus dem ScatterPlot herausgelesen werden. Um dennoch an alle Details zu den einzelnen Suchtreffern zu gelangen, wird das Konzept von *Brushing and Linking* [BC87] verwendet, um vom ScatterPlot aus durch das „Fernsteuern“ der HyperGrid die Detailinformationen in dieser sichtbar zu machen. Während durch einen einfachen Mouseover-Effekt beim Überfahren eines Medientyp-Symbols im ScatterPlot der Titel des besagten Datenobjekts

und dessen Erscheinungsjahr als Tooltip sichtbar wird (siehe Abbildung 19, im unteren linken Bereich beim Film Duell), wird durch Klicken auf das Icon ein Pie-Menü [HCW88] geöffnet (siehe Abbildung 20).

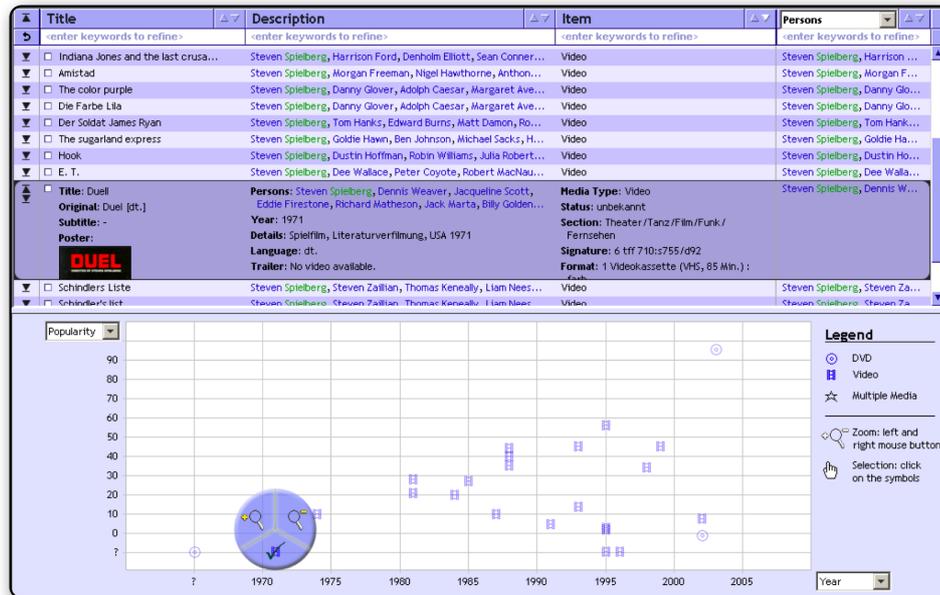


Abbildung 20: Die HyperGrid lässt sich vom ScatterPlot aus durch ein Pie-Menü „fernsteuern“.

Mit dem Pie-Menü ist es dem Benutzer möglich, drei verschiedene Aktionen auszuführen. Mit einem Klick auf das Lupensymbol mit einem Plus (links) wird in der HyperGrid eine Zoom-In-Aktion auf dem entsprechenden Datenobjekt ausgeführt. Durch einen Klick auf das Lupensymbol mit dem Minuszeichen (rechts) wird entsprechend wieder aus den Detailinformationen des Datenobjekt heraus gezoomt. Als dritte Option bietet das Pie-Menü dem Benutzer an, das Datenobjekt direkt aus dem ScatterPlot heraus zu selektieren.

Damit der Benutzer alle diese Effekte und Möglichkeiten wahrnimmt, wird jedes Mal, wenn mit der Maus im ScatterPlot ein Medientyp-Symbol überfahren wird, auch der Fokus in der HyperGrid auf das entsprechende Datenobjekt gesetzt. Dadurch wird dieses automatisch in den sichtbaren Bereich der HyperGrid gescrollt. Umgekehrt wird auch beim Überfahren einer HyperGrid-Zeile mit dem Maus-Cursor das Objekt im ScatterPlot hervorgehoben. Damit wird auch über die HyperGrid die Position eines bestimmten Datenobjekts innerhalb der Übersichtsdarstellung zugänglich gemacht. Auf Grund dieser indirekten Verbindung der beiden Visualisierungen durch das Konzept von Brushing and

Linking entsteht für den Benutzer allerdings ein kognitiver Aufwand. Ausserdem können wie im Anwendungsszenario (Kapitel 4.1) gezeigt Platzprobleme auftreten, wenn mehrere Visualisierungen parallel auf dem Bildschirm angezeigt werden müssen.

5 HyperScatter

5.1 Ziele des HyperScatters

Mit dem HyperScatter soll ein leistungsstarkes Visualisierungs- und Analyse-Werkzeug entwickelt werden, welches im Rahmen von visuellen Suchsystemen zum Einsatz kommt. Dabei sollen vor allem auch unerfahrene Nutzer in die Lage versetzt werden umfassende Analysen auf dem Datenbestand des Suchsystems durchführen zu können. Um dieser Anforderung gerecht werden zu können, setzt der HyperScatter dabei auf zwei wesentliche Konzepte: Das Zoomable User Interface und die Scatterplot-Darstellung.

5.1.1 Gründe für eine Scatterplot-Visualisierung

Im Gegensatz zu gewöhnlichen listen- und tabellenbasierten Darstellungen von Suchtreffermengen, können mit einer Scatterplot-Visualisierung theoretisch beliebig grosse Mengen an Datenpunkten auf einen Blick dargestellt werden (praktisch muss dabei allerdings das Problem des Overplottings, also des sich gegenseitigen Überdeckens von Datenpunkten, mitberücksichtigt werden). Im Fall vom HyperScatter wird der Visualisierung dabei ein zwei-dimensionaler Scatterplot zu Grunde gelegt. Das bedeutet, dass sich auf einen Blick alle im Resultatset befindlichen Datenobjekte anhand von zwei Dimensionen vergleichen lassen. So können durch die visuelle Analyse des Scatterplots schnell (für den Nutzer) relevante Datenobjekte von den weniger wichtigen isoliert werden, was in den meisten Suchsystemen eine wesentliche Aufgabe darstellt. Die Aufteilung der Resultatmenge in interessante und eher uninteressante Bereiche, kann dabei alleine durch die Betrachtung der Visualisierung und der Anordnung der Datenpunkte darauf vollzogen werden.

Ferner ist es mit einem Scatterplot möglich, nicht nur einzelne interessante Datenobjekte aus der Treffermenge zu extrahieren, mit einer Visualisierung wie dem HyperScatter kann auch die gesamte Treffermenge an sich analysiert werden. Wenn man beispielsweise in einer Filmdatenbank eine Suchanfrage nach *Steven Spielberg* abschickt und sich das Suchresultat in einer scatterplot-artigen Visualisierung darstellen lässt (siehe Abbildung 21), wobei auf die Y-Achse die Beliebtheit und auf die X-Achse das Erscheinungsjahr der Filme abgetragen werden soll, so können schnell Aussagen über die gesamte Treffermenge gemacht werden. Aus Abbildung 21 ist beispielsweise zu lesen, dass Steven Spielbergs

erster Film (der in dieser Datenbank hinterlegt ist) kurz nach 1970 erschienen ist und dass es wohl in der Mediothek im Moment „einen“ Film von Steven Spielberg gibt, der sich einer besonderen Beliebtheit erfreut. Dieser Film wird durch den Datenpunkt rechts oben im Scatterplot von Abbildung 21 repräsentiert. Je weiter oben ein Datenpunkt in der Abbildung platziert ist, desto höher ist seine Popularitäts-Wertung. Das Anzeigen vieler solcher Gegebenheiten auf einen Blick ist ein Mehrwert, der von den meisten Visualisierungen, welche sich auf die Analyse von Detailinformationen zu einzelnen Datenobjekten spezialisiert haben, nicht geboten werden kann.

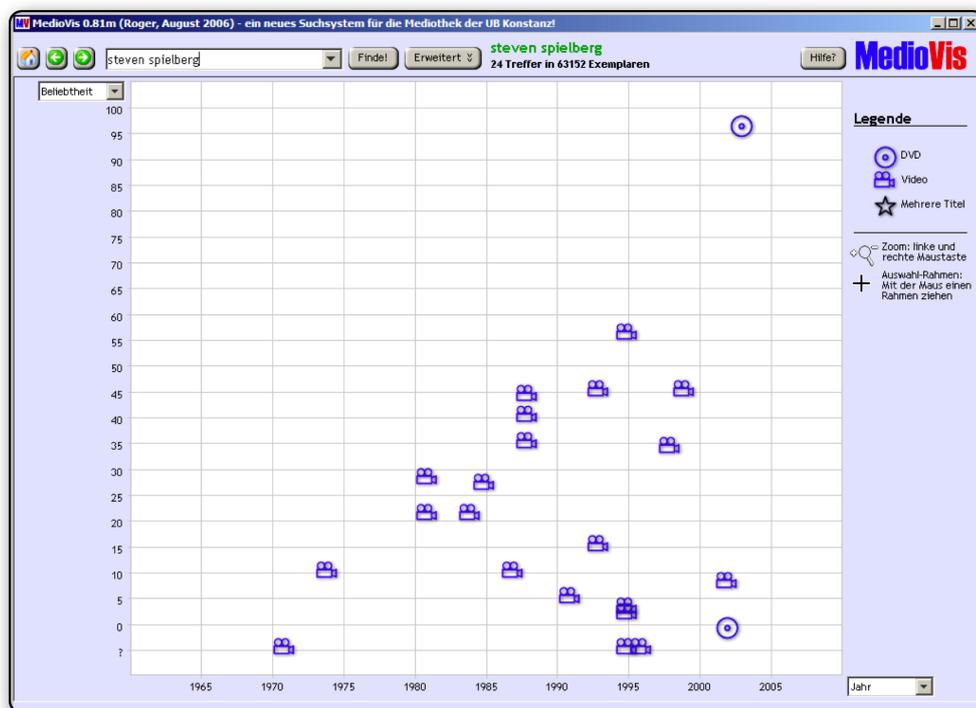


Abbildung 21: Der HyperScatter in MedioVis: Trefferdarstellung zur Suchanfrage *Steven Spielberg*.

Ein weiterer Vorteil von Scatterplot-Visualisierungen ist, dass durch das graphische Verorten der Datenpunkten in der Visualisierung, die Beziehungen zwischen den Objekten klar dargestellt werden können. Wenn man sich zum Beispiel eine Zeitachse wie die des Scatterplots in Abbildung 21 anschaut, so sind da die Abhängigkeiten der verschiedenen Filme bezüglich ihres Erscheinungsjahres leicht erkennbar. Stellt man hingegen das gleiche Set an Filmen zum Beispiel in einer Tabelle dar, in der jede Zeile einem Film und die Spalten jeweils einem Attribute des Filmes entspricht (in einer Spalte davon liegt das Erschei-

nungsjahr, nach dem die Tabelle auch sortiert ist), treten verschiedene Effekte auf, die bei der Interpretation der Tabelle beim Benutzer zu Missverständnissen führen können. Zum einen müssen Filme die im genau gleichen Jahr erschienen sind trotzdem untereinander, also in einer Reihenfolge, in der Tabelle dargestellt werden, was beim Benutzer unter Umständen das falsche Verständnis hervorrufen kann, dass der obere Film vor, bzw. je nach Sortierreihenfolge nach dem unmittelbar darunter stehenden Film in der Tabelle erschienen ist. Auch werden in dieser Tabelle alle Filme in regelmässigem Abstand untereinander abgetragen, womit beim groben Hinsehen die Information über zeitliche Lücken zwischen den Erscheinungsjahren der Filme bzw. über Häufungen von Erscheinungen in bestimmten Jahren verloren geht.

5.1.2 Gründe für ein Zoomable User Interface

Beim Zoomable User Interface handelte es sich um ein Interaktionskonzept, welches durch eine starke Analogie zur physischen Navigation im Raum geprägt ist. Durch dieses intuitive Navigationskonzept kann vom Benutzer ein verständliches mentales Modell der Visualisierung aufgebaut werden. Um dies zu unterstützen werden Übergänge zwischen verschiedenen Zustände einer Visualisierung als flüssig animierte Zoombewegungen visualisiert. Das Umsetzen dieses Konzepts stellt die zweite wichtige Komponente des HyperScatters dar.

In herkömmlichen Suchsystemen, wie beispielsweise bei den Websuchmaschinen von Google und Yahoo, wird der Benutzer stark mit der Aufgabe des Fenstermanagements belastet, weil, um von einer Suchtrefferdarstellung zu den einzelnen Dokumenten vorzudringen, oft ein neues Bildschirmfenster geöffnet werden muss. Das wirkt sich vor allem dann negativ aus, wenn der Benutzer zuerst mehrere Objekte aus einer Treffermenge genauer untersuchen muss, bevor er schliesslich, über das Vergleichen ihrer Detailinformationen, die für ihn relevanten Datenobjekte bestimmen kann. Im Extremfall besteht sogar die Gefahr, dass der Anwender den Kontext, also die Verbindung zur eigentlichen Aufgabenstellung verliert, da er nach einiger Zeit nicht mehr nachvollziehen kann welche Bildschirmfenster mit Detailinformationen zu welcher Suchanfrage und zu welchem konkreten Suchtreffer gehören. Dieser kognitive Aufwand kann unter Verwendung eines Zoomable-User-Interface-Ansatzes vermieden werden, indem beides, Übersicht und Detailinformationen, direkt in der gleichen Visualisierung und somit im gleichen Bildschirmfenster angezeigt und durch Zooming zugänglich gemacht wird [PF93]. Somit verliert der Anwender auch beim genaue-

ren Betrachten von Detailinformationen zu bestimmten Datenobjekten den Kontext der getätigten Suchanfrage und deren Treffermenge nicht.

5.2 Interaktionskomponenten des HyperScatters



Abbildung 22: Die verschiedenen Komponenten von MedioVis und dem HyperScatter.

Abbildung 22 zeigt den HyperScatter, der hier als Visualisierung in das MedioVis-Framework eingebaut ist. Im oberen Bereich (1) sind die Interaktionskomponenten von MedioVis zu sehen. Auf diesen Bereich soll in dieser Arbeit nicht weiter eingegangen werden, da dies nicht direkt mit der HyperScatter-Visualisierung in Zusammenhang steht. Alle diese Interaktionskomponenten werden an dieser Stelle angezeigt, ganz egal was für Visualisierungen gerade in MedioVis verwendet werden.

Der Bereich (2) ist die eigentliche Scatterplot-Komponente des HyperScatters. Hier werden die verschiedenen Datenobjekte, die als Treffer einer Suchanfrage von MedioVis zurückgeliefert werden anhand der beiden Achsen (3) Y-Achse und (4) X-Achse im Koordi-

natensystem verortet und mit einem ihrem Medientyp entsprechenden Symbol visualisiert. Eine Legende, welche die verschiedenen Symbole erklärt, befindet sich im Bereich (5).

5.3 Visualisierungs- und Interaktionskonzept des HyperScatters

Im folgenden Kapitel soll nun konkret das Visualisierungs- und Interaktionskonzept des HyperScatters präsentiert werden. Dies soll anhand des Visual Information Seeking Mantras von Shneiderman [Shn96] in eine Struktur gebracht werden.

Shneiderman stellt zwei Arbeitsweisen bei der Benutzung von visuellen Suchsystemen heraus. Im einen Fall, der Known-Item Search, hat ein Benutzer eine ganz klare Vorstellung, nach was er im gegebenen Informationsraum sucht. Er kennt das genaue Resultat seiner Suche und ist darum auch in der Lage, geeignete Suchanfragen zu formulieren. Im anderen Fall, dem Browsing-Ansatz, ist der Anwender nicht in der Lage sein Informationsbedürfnis in eine klare Suchanfrage, zum Beispiel über die Eingabe von Schlüsselwörtern in eine Suchmaske, zu formulieren. Er muss oder möchte sich über das mehr oder weniger freie Bewegen im Informationsraum über sein Informationsbedürfnis klar werden und dieses dabei zugleich auch stillen. Um ein umfänglich durchdachtes visuelles Suchsystem darzustellen, sollten beide dieser Vorgehensweisen unterstützt werden. Dazu hat Shneiderman sieben Kernaufgaben identifiziert, welche durch ein visuelles Suchsystem abgedeckt werden sollen: Overview, Relate, Zoom, Detail-On-Demand, Filter, History und Extract.

5.3.1 Overview

Bei der Kernaufgabe *Overview* geht es darum, dass der Anwender eine Übersicht über eine Kollektion von Datenobjekten erhält. In einem Informationssuchsystem wie MedioVis würde dies zum Beispiel der Übersicht über die gesamte Treffermenge einer Suchanfrage entsprechen. Dem Anwender sollte es jederzeit möglich sein, zu dieser Übersicht zu gelangen, um von da aus in die Detailinformationen der Treffermenge einzutauchen.

5.3.1.1 Übersichtsdarstellung mit dem HyperScatter

Wie schon erwähnt zeichnen sich Scatterplot-Visualisierungen durch die Fähigkeit aus, eine gute Übersicht über einen Datenraum darstellen zu können. Auf einer begrenzten Bildschirmfläche können im Prinzip beliebig viele Datenobjekte dargestellt werden. Diese werden als Punkte auf dem Bildschirm verortet (siehe Abbildung 23). Somit kann die gesamte Suchtreffermenge auf einen Blick dargestellt werden, was bei listen- oder tabellebasierten Trefferdarstellungen im allgemeinen nicht möglich ist. Bei diesen muss der Benutzer meistens vertikal durch die Treffermenge scrollen, oder sogar durch mehrere Bildschirmseiten blättern, um Schritt für Schritt einmal an alle Suchtreffer zu gelangen. Im HyperScatter werden die einzelnen Datenobjekte nicht nur als Punkte bzw. Pixel dargestellt, sondern es wird je nach Medientyp des Datenobjekts (also DVD, CD, Buch etc.) ein entsprechendes Symbol im Scatterplot angezeigt.

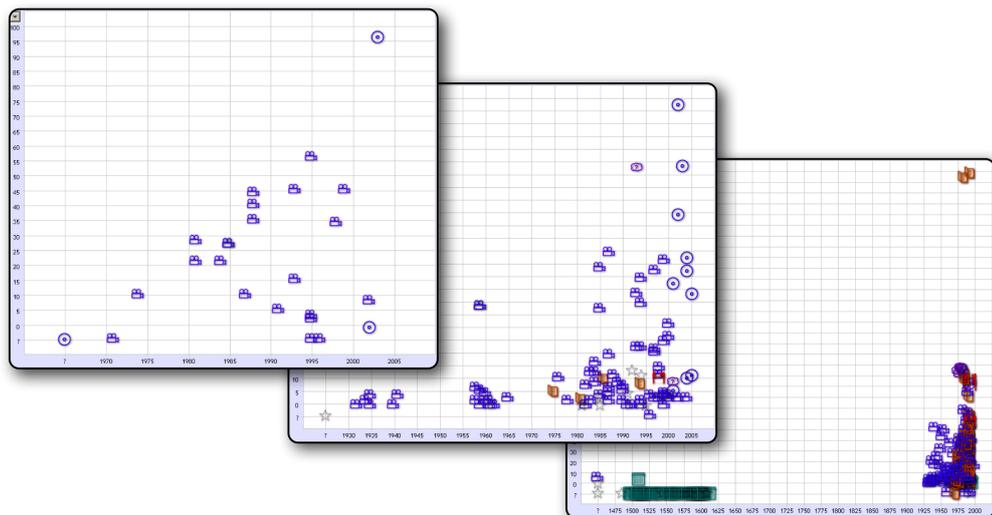


Abbildung 23: Im HyperScatter können theoretisch beliebig grosse Menge an Datenpunkten angezeigt werden: (Von links nach rechts) HyperScatter mit 24 Datenpunkten — mit 123 Datenpunkten — mit 1000 Datenpunkten.

Aus der Abbildung wird auch klar, dass Überlappungs-Probleme entstehen, je mehr Datenpunkte auf einmal dargestellt werden. Diese Probleme gilt es mit dem HyperScatter zu lösen. Einerseits steigt die Anzahl von Datenpunkten an, die genau auf dem gleichen Punkt im Scatterplot zu liegen kommen (weil sie in den beiden auf die Achsen gemapten Attributen exakt die gleichen Werte aufweisen). Diese Punkte werden im HyperScatter zu Multiple Datapoints zusammengefasst und in der Übersichtsdarstellung zuerst einmal wie

einzelne Datenpunkte behandelt. Andererseits häuft sich auch die Wahrscheinlichkeit der Anzahl von Datenpunkten, die zwar nicht exakt an derselben Stelle, aber sehr nahe bei einander zu liegen kommen und so Cluster von Datenpunkten bilden, welche sich teilweise überdecken. Dabei tritt auch oft der Fall auf, dass weiter unten liegende Datenpunkte vom Knäuel der darüber liegenden Datenobjekte vollkommen verdeckt werden und überhaupt nicht mehr für eine direkte Interaktion zugänglich sind.

Die Bildung von solchen Datapoint Clustern wird ausserdem verstärkt, wenn auf den Scatterplot-Achsen grosse Wertebereiche abgetragen werden müssen, weil es zum Beispiel, wie in Abbildung 23 rechts, eine Gruppe von Datenobjekten mit einer sehr frühen Altersangabe und eine Gruppe von Datenobjekten mit einem eher jungen Erscheinungsdatum gibt. Im Beispiel muss darum die gesamte Zeitspanne von 1475 bis heute abgetragen werden, was dazu führt, dass ein grosser Bereich (1625 bis 1925) des Scatterplots ungenutzt bleibt. Man könnte argumentieren, dass die Zeitachse in diesem Bereich verzerrt werden sollte, damit für die beiden Datapoint Cluster mehr Platz zur Verfügung steht, um diese etwas aufzulockern. Es gibt aber zwei gewichtige Gegenargumente, weshalb bei der Entwicklung des HyperScatter gegen diese Vorgehensweise entschieden worden ist.

Zum ersten lassen sich viele Benutzer bereits durch logarithmisch skalierte Scatterplot-Achsen in die Irre führen, weil diese ein nicht wahrheitsgetreues Bild darstellen. Ein Anwender erwartet eine lineare Zuordnung gegenseitiger Abstände einzelner Datenpunkte in einem Scatterplot. Eine logarithmische Achsenbelegungen würde also ein nicht erwartungskonformes Bild erzeugen und könnten zur falschen Interpretation der Visualisierung führen. Im betrachteten Fall wäre der Effekt der Verzerrung des Scatterplots noch viel komplizierter nachzuvollziehen, da die Achsen nicht nur einfach logarithmisch skaliert, sondern an mehreren Stellen auf der Achse verzerrt werden müsste. Vom Usability-Standpunkt her ist ein solches Verhalten, bei einer Visualisierung die für den Nicht-Experten konzipiert wird, nicht vertretbar.

Zum Zweiten ist es aber auch so, dass bei einer Scatterplot-Visualisierung die Information nicht nur alleine in den aufgetragenen Datenpunkten liegt. Auch die Zwischenräume und die leeren Bereiche der Visualisierung sind Informationsträger. Im Beispiel von Abbildung 23 rechts kann sofort herausgelesen werden, dass durch die Suchanfrage, welche zu dieser Trefferdarstellung geführt hat, einerseits viele sehr alte Dokumente in der Mediothek gefunden wurden (es handelt sich in diesem Beispiel vor allem um Mikrofilmaufnahmen von alten Schriften, welche ebenfalls in der Mediothek aufbewahrt werden), andererseits

aber auch eine grosse Menge an Filmen und Büchern aus neuerer Zeit auf die Anfrage angesprochen hat, dazwischen aber entweder überhaupt keine Dokumente in der Mediothek vorhanden sind, oder aber in dieser Zeitspanne die Inhalte der Suchanfrage nicht thematisiert worden sind.

5.3.1.2 Aktualisierung der beiden Achsen und der Legende des HyperScatters

Im HyperScatter wird der Wertebereich der beiden Achsen immer automatisch an die im Scatterplot dargestellten Datenpunkte angepasst. Das heisst, wenn durch das Abschicken einer neuen Suchanfrage der Inhalt des Scatterplots verändert wird, werden die beiden Achsen neu berechnet, so dass der zur Darstellung des Scatterplots verwendete Bildschirmplatz stets optimal ausgenutzt wird. Wenn die Suchtreffer in die Visualisierung eingelesen werden, wird jeweils der Maximalwert und der Minimalwert für die beiden auf die Achsen gemapten Attribute aus allen Suchtreffern bestimmt und damit der Wertebereich der beiden Achsen festgelegt (siehe Abbildung 24).

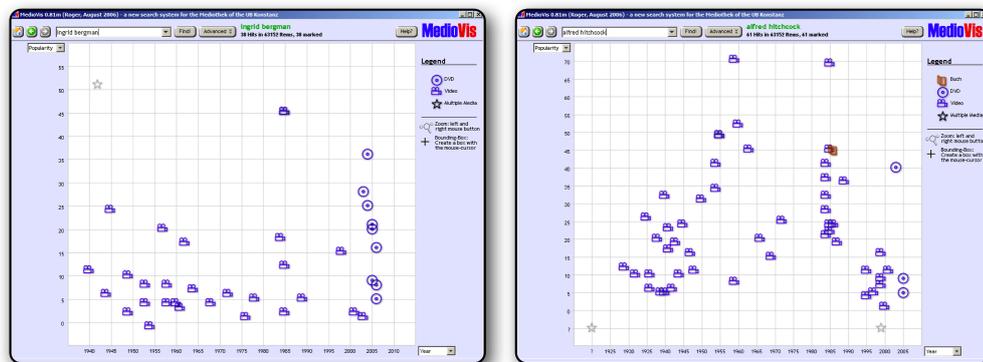


Abbildung 24: Im HyperScatter werden die Wertebereiche der Achsen und die Legende automatisch an die Treffermenge angepasst.

Auch die Legende wird nach jeder Suche automatisch neu berechnet. Dadurch sind in der Legende immer nur die Medientyp-Symbole erklärt, von denen sich auch tatsächlich Datenobjekte in der aktuellen Treffermenge befinden. In der Treffermenge zur Suchanfrage *Ingrid Bergman* (links) sind in der Mediothek zum Beispiel nur DVDs und Videokassetten vorhanden. Zur Anfrage nach *Alfred Hitchcock* (rechts) gibt es neben DVDs und Videokassetten auch noch ein Buch.

5.3.1.3 Technische Umsetzung der Verortung von Datenpunkten im HyperScatter

An dieser Stelle soll kurz auf einen Aspekt der technischen Umsetzung des HyperScatters, der Verortung der Datenpunkte auf dem Scatterplot, eingegangen werden. MedioVis ist im Wesentlichen nach dem Model-View-Controller-Pattern [Gol84] aufgebaut. Für die interne Datenhaltung der Datenobjekte, welche als einzelne Suchtreffer von der Datenbankanfrage zurückgeliefert werden, gibt es eine Klasse namens DocManager, welche im Wesentlichen ein Array von einzelnen Document-Instanzen enthält. Einem Document ist entsprechend ein einzelner Suchtreffer, zum Beispiel eine in der Mediothek vorhandene DVD, zugeordnet. Bei einer Suchanfrage werden also die Detailinformationen zu den einzelnen Suchtreffern jeweils in einem Document abgespeichert und alle Document-Objekte einer Suchanfrage im DocManager abgelegt. Im DocManager wird zusätzlich jedem Document eine ID zugeordnet, über welche danach auf das Document zugegriffen werden kann. Im Folgenden soll aber zur Vereinfachung angenommen werden, dass beim Zeichnen des HyperScatters alle Datenobjekte (Documents) direkt in der Visualisierung umgesetzt werden.

Abbildung 25 zeigt das schematische Vorgehen bei der Verortung der einzelnen Datenpunkte im HyperScatter. Dabei stellt die Klasse HyperScatterMatrix das eigentliche Panel dar, auf dem der Inhalt des Scatterplots gezeichnet wird. XAxis und YAxis sind die beiden Klassen, die für die X- und die Y-Achse des HyperScatter verantwortlich sind. Einerseits beinhalten sie je ein Panel, auf dem wirklich die Achse gezeichnet wird, andererseits implementieren sie auch die Methode getPosition(Document). Dieser Methode kann ein Document übergeben werden und es wird die Position dieses Documents in Pixeln anhand der im Document gespeicherten Attributwerte berechnet und zurückgegeben. Dazu müssen natürlich die beiden Klassen XAxis und YAxis wissen, welches Attribut im Moment gerade auf ihnen abgetragen ist. Wenn also auf der X-Achse des HyperScatters das Erscheinungsjahr angezeigt wird und die Methode getPosition(Document) mit einem bestimmten Datenobjekt aufgerufen wird, holt sich die Klasse XAxis den Erscheinungsjahr-Wert aus besagtem Document und berechnet daraus die X-Koordinate des Datenpunktes auf dem Scatterplot in Pixeln. Analog dazu verhält es sich mit der Y-Achse. In der Klasse Datapoints werden schliesslich die einzelnen Positionen mit den Document-Objekten zusammen gespeichert, damit nicht bei jedem Aufruf von repaint() alle Positionen neu berechnet werden müssen, sondern nur dann, wenn sich wirklich etwas an den Positionen der Datenpunkte ändert bzw. durch eine neue Suchanfrage komplet neue Datenobjekt in den HyperScatter geladen werden.

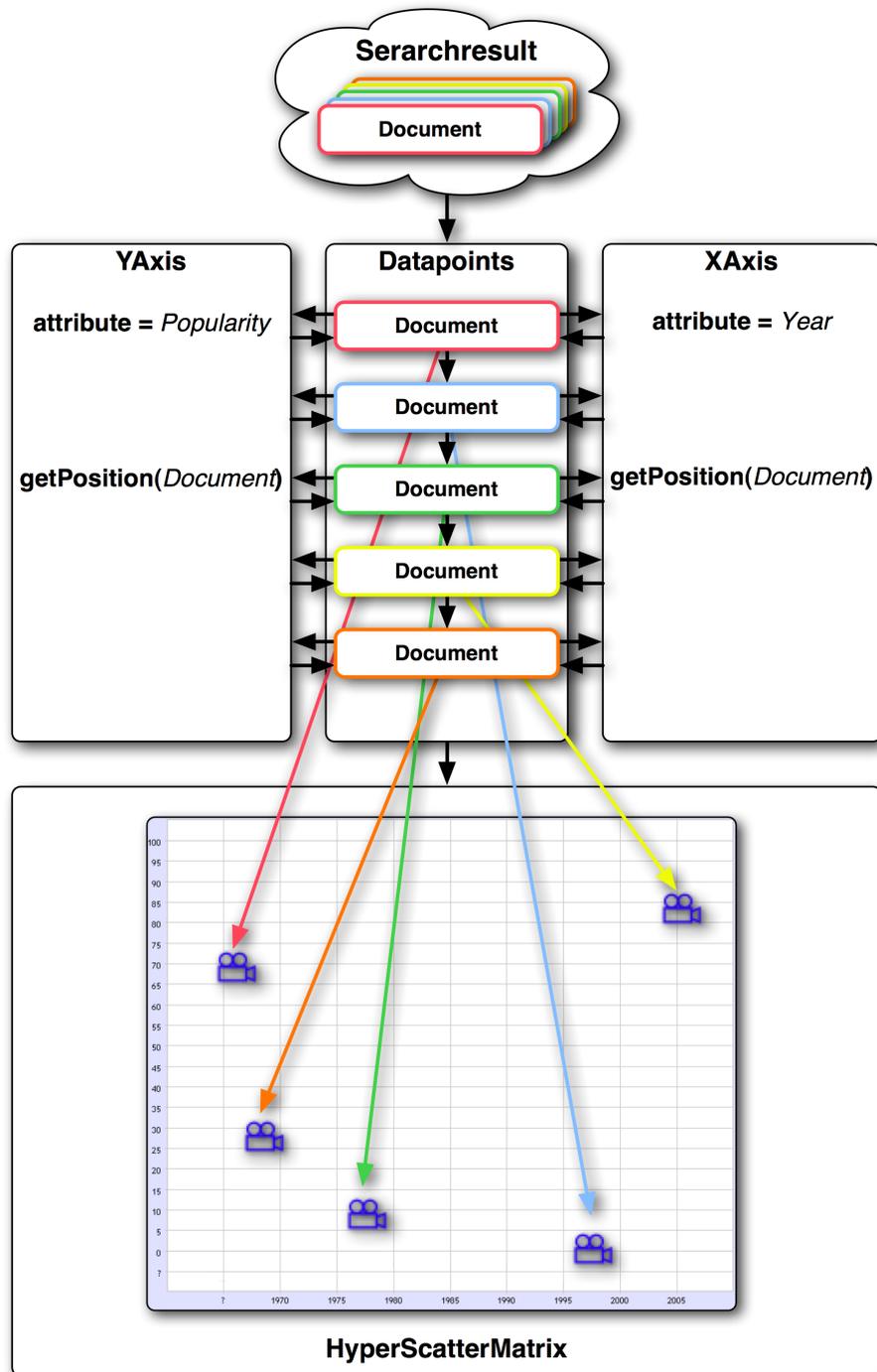


Abbildung 25: Schematische Darstellung wie nach einer neuen Suchanfrage an MedioVis die Treffer-Datenpunkte im HyperScatter verortet werden. Dabei sind folgende Java-Klassen involviert: Datapoints, XAxis, YAxis und HyperScatterMatrix.

5.3.2 **Relate**

Eine weitere Kernaufgabe von visuellen Suchsystemen ist es, Zusammenhänge zwischen den einzelnen Datenobjekten darzustellen. Damit können alle Elemente, die einem bestimmten Pattern folgen, welches sich durch mehrere verschiedene Attributwerte definieren kann, ausfindig gemacht werden. Zum Sichtbarmachen von Relationen zwischen einzelnen Datenobjekten genügen manchmal schon einfache Mechanismen wie das Sortieren in einer Tabelle nach einer einzelnen Spalte. Auf diese Weise ist es zum Beispiel in der HyperGrid möglich Datenobjekte aufzuspüren, die ein ähnliches Erscheinungsjahr aufweisen. Dazu wird, das Erscheinungsjahr in der User Adjustable Column aus der Dropdown-Liste auszuwählen um danach nach dieser Spalte zu sortieren. Nun ist es relativ einfach, aus der sortierten HyperGrid-Tabelle die Datenobjekte zu erkennen, die ein nahe beieinander liegendes Erscheinungsjahr aufweisen.

Noch verständlicher wird der Zusammenhang von bestimmten Datenobjekten durch eine graphische Darstellung ersichtlich. Das Erscheinungsjahr von Filmen kann beispielsweise durch eine Zeitachse visualisiert werden. Dabei wird die Stärke der Relation zwischen zwei Datenobjekten direkt durch deren Abstand zu einander ausgedrückt.

5.3.2.1 **Visualisierung von Zusammenhängen mit dem HyperScatter**

Der HyperScatter realisiert eine solche graphische Darstellung und ermöglicht es dadurch Zusammenhänge zwischen einzelnen Datenpunkten klar darzustellen [Tuf83]. Wenn man zum Beispiel auf eine Achse des HyperScatters das Erscheinungsjahr abgetragen hat, erhält man genau die oben erwähnte Zeitachse.

Da im HyperScatter aber zwei Achsen vorhanden sind, können Relationen zwischen den Datenobjekten einer Treffermenge noch weitreichender analysiert werden. Wenn beispielsweise auf eine Achse des HyperScatters das Erscheinungsjahr und auf die andere die Popularität der Datenobjekte gemapt wird (wie dies in Abbildung 21 der Fall ist) können leicht die neuen und populären Filme oder die unpopulären Filme der 90er Jahre aus den angezeigten Datenobjekten ausfindig gemacht werden. Damit lassen sich schon recht komplizierte Zusammenhänge zwischen den Datenobjekten einer Suchtreffermenge relativ einfach darstellen.

5.3.2.2 Medientyp-Symbole als dritte Dimension

Grundsätzlich lassen sich mit dem HyperScatter durch seine zwei Achsen Datenobjekte anhand von zwei in Relation gesetzten Dimensionen auf der Visualisierungsfläche darstellen. Bei der Visualisierung der Mediodaten kann aber über die Codierung des Medientyps durch ein passendes Symbol, welches anstelle eines einfachen Punktes im Scatterplot-Bereich der Visualisierung dargestellt wird, eine weitere Dimension direkt im HyperScatter verankert werden. Dies wird durch die überschaubare Anzahl, von neun verschiedenen Medientypen in der Mediodatenbank unterstützt (siehe Abbildung 26).



Abbildung 26: Im HyperScatter werden die verschiedenen Medientypen durch folgende Symbole repräsentiert (von links nach rechts): *Videokassette*, *DVD*, *CD*, *Tonträger*, *Buch*, *Zeitschrift*, *Karte*, *Mikrofilm* und *Diskette*. Dazu kommt das Symbol für *Diverses* und für *Multiple Datapoints* in der unteren Reihe.

Wenn es noch mehr Medientypen geben würde, müsste untersucht werden, ob das Darstellen dieser dritten Dimension im HyperScatter wirklich von Vorteil ist oder ob es eher zu Verwirrung führt, in dem es die visuelle Darstellung der Treffermengen unruhig erscheinen lässt und der HyperScatter durch einen Informationsüberfluss überladen wird. Dabei muss aber auch in Betracht gezogen werden, dass im Allgemeinen bei einer Suchanfrage immer nur eine Auswahl an Medientypen in der Treffermenge vorkommt, was die Problematik etwas entschärft.

5.3.2.3 Variable Achsenbelegung im HyperScatter

Eine weitere Besonderheit des HyperScatters, welche auch schon Bestandteil des Scatter-Plots aus dem klassischen Anwendungsszenario war, ist, dass die Achsenbelegungen durch den MedioVis-Benutzer frei wählbar sind. Durch die Auswahl mittels einer Dropdown-Liste (je eine pro Achse) kann der Anwender eines der vordefinierten Attribute auf die Achse abtragen lassen und so zum Beispiel statt der Popularität die Sprache auf die Y-Achse legen. Die Visualisierung wird dabei sofort nach der Neubelegung einer Achse aktualisiert

und alle Datenpunkte werden neu auf dem HyperScatter verortet, wie es in Abbildung 27 veranschaulicht wird.

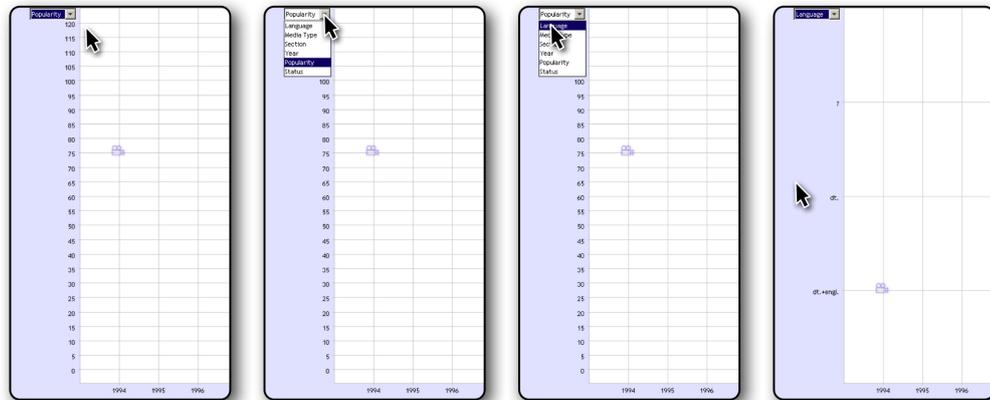


Abbildung 27: Durch eine Dropdown-Liste, welche für beide Achsen des HyperScatters zur Verfügung steht, kann der Benutzer das auf die Achse gemapte Attribut jederzeit neu festlegen.

Im HyperScatter ist es möglich, sowohl metrische und ordinale als auch nominale Attribute auf die beiden Achsen zu legen. Obwohl dies immer wieder als kritisch angesehen wird, wurde bei der Konzeption des HyperScatters nach der Überzeugung gehandelt, dass das Abtragen von nominalen Daten auf eine Scatterplot-Achse durchaus einen wichtigen Informationsgewinn beim Anwender bewirken kann. Dies wurde schon bei einigen Projekten der *Arbeitsgruppe Human-Computer Interaction*, wie dem ZuiScat [BR05] und VisMeB aber auch bei Projekten anderer Forschergruppen [SFRG00] und [MHML05] festgestellt. Als Beispiel für das Mediotheksszenario könnte man sich vorstellen, dass ein Benutzer, wie in Abbildung 27 angedeutet wird, das nominale Attribut *Sprache* auf die Y-Achse des HyperScatters legt, und so schnell alle Filme mit deutscher Vertonung aus der Treffermenge herauslesen kann.

5.3.3 Details-On-Demand

Das Nutzerbedürfnis, im HyperScatter Detailinformationen eines einzelnen Datenobjekts angezeigt zu bekommen, ist leicht nachzuvollziehen. In einer Scatterplot-Visualisierung können nur so viele Attribute der Datenobjekte dargestellt werden, wie der Scatterplot Achsen hat. Dazu kommen noch die Attribute, welche durch die Farbe und die Form der Datenpunkte codiert werden. Im Fall des HyperScatters sind dies also drei Attribute pro

Objekt, zum Beispiel das Erscheinungsjahr und die Popularität eines Filmes und dazu der Medientyp welcher über das Medientyp-Symbol codiert wird (siehe Abbildung 21). Im Rahmen von MedioVis soll davon ausgegangen werden, dass sich ein Anwender nicht nur für drei Attribute interessiert, sondern dass er, sobald ein bestimmtes, relevantes Datenobjekt identifiziert wurde, zusätzliche Detailinformationen wie Titel, Ausleihstatus, Standort in der Mediothek etc., angezeigt bekommen will. Im Szenario zur klassischen Benutzung von MedioVis musste er dafür im ScatterPlot den „Umweg“ über die HyperGrid machen. Wenn die verwendete Visualisierung, hier also der HyperScatter, die Funktionalität des semantischen Zoomings selber zur Verfügung stellt, wird dem Benutzer die Arbeit erleichtert.

Im Visual Information Seeking Mantra wird dabei als generelle Herangehensweise vorgeschlagen, diese Detailinformationen bei Bedarf mittels eines Pop-Up-Fensters, welches durch Klicken auf ein Datenelement erscheint, darzustellen. Dies wurde zum Beispiel im FilmFinder [AS94b] so umgesetzt. Pop-Up-Fenster bringen allerdings den Nachteile von erhöhtem kognitivem Aufwand für den Benutzer [KS96] und der Gefahr von Kontextverlust des Anwenders mit sich. Im HyperScatter wird deshalb, durch Anwenden des Zoomable-User-Interface-Konzepts, auf solche Pop-Up-Fenster verzichtet und die Detailinformationen werden durch Heranzoomen der Informationen, direkt im Kontext der Trefferdarstellung angezeigt. Obwohl hier ein Zooming-Ansatz gewählt wird, soll darauf hingewiesen werden, dass dies nichts mit der von Shneiderman definierten Kernaufgabe *Zoom* zu tun hat, welche im Abschnitt 5.3.4 beschrieben wird.

5.3.3.1 Geometrischer vs. semantischer Zoom

Wie im Kapitel 2 ausgeführt wird, gibt es mehrere Arten von Zooming, welche sich aber meistens jeweils als eher geometrisch- oder eher semantisch-getriebenes Zooming einstufen lassen. In vielen Fällen ist dabei die geometrische Komponente relativ stark ausgeprägt, was sicher darauf zurück zu führen ist, dass das geometrische Zooming sehr intuitiv zu verstehen ist. Beim HyperScatter wird im Gegensatz dazu ein stark semantisch geprägter Zoom eingesetzt. Dies hat zwei Hauptgründe, welche hier erläutert werden sollen.

Es ist ein erklärtes Ziel der HyperScatter-Visualisierung, eine gute Übersichtsdarstellung über die gesamte Treffermenge einer Suchanfrage zu liefern. Ein geometrisches Zooming im HyperScatter führt aber dazu, dass diese Übersicht nicht mehr vorhanden ist, da nur noch

ein Teilbereich der Treffermenge zu sehen ist. Dies kann zwar durchaus auch Vorteile haben, wie im Abschnitt 5.3.4 gezeigt wird und wird im HyperScatter sogar auch eingesetzt. Allerdings nicht für die Aufgabe des Detail-On-Demand-Zoomings.

Auch ist es für die Darstellung der Detailinformationen von einzelnen Datenobjekten im HyperScatter nicht sinnvoll, geometrisches Zooming zu verwenden, da lediglich der Datenpunkt selber genauer Betrachtet werden soll. Der ihn umgebende Bereich im Scatterplot hat, in Bezug auf Detailinformationen zu besagtem Datenobjekt, wenig bis gar keine Aussagekraft. Dies wäre anders, wenn es sich nicht um einzelne Datenpunkte, sondern um eine kontinuierlich gefüllte Datenlandschaft, wie beispielsweise einer Landkarte, handeln würde.

5.3.3.2 Detail-Zoom im HyperScatter

Das Heranzoomen der Detailinformationen von einzelnen Datenobjekten erfolgt im HyperScatter in drei fest vordefinierten Stufen (siehe Abbildung 29), bei Multiple Datapoints gibt es vier Stufen (siehe Abbildung 30). Dieses semi-automatische Zoomen mit vorgegebenen Zoomstufen soll dem Benutzer den kognitiven Aufwand ersparen, die richtige Skalierung für sein Informationsbedürfnis finden zu müssen und so das Arbeiten mit dem System erleichtern [Ras00]. Das Einstellen eines geeigneten Skalierungsfaktors ist vor allem bei textuellen Daten recht schwierig und aufwändig, da einerseits möglichst viel Text dargestellt werden soll, jedoch eine Schriftgrösse gefunden werden muss, die das bequeme Ablesen der Informationen auf dem Computerbildschirm ermöglicht. Ausserdem wird durch die Wahl von wenigen Zoomstufen und dem daraus resultierenden kleinen Interaktionsaufwand für den Benutzer das Arbeiten mit dem HyperScatter vereinfacht und beschleunigt.

Das Hineinzoomen in die Detailansicht und das Wieder-Herauszoomen wird dabei so animiert, dass der Benutzer das Gefühl vermittelt bekommt, ein Datenpunkt werde „aufgeblasen“. Es ändert sich also nur die Repräsentation eines Datenpunktes, der im Scatterplot von einem Punkt zu einem Panel anwächst, auf dem die Detailinformationen des entsprechenden Datenobjekts angezeigt werden. Der darunterliegende Scatterplot bleibt dabei wie erwähnt unverändert (siehe Abbildung 29).

Es ist klar, dass durch dieses Detail-Panel eines Datenobjekts ein grosser Teil des HyperScatters verdeckt wird. Darum wird der Hintergrund von den Detail-Panels halb-

transparent gezeichnet, so dass der darunterliegende Scatterplot durch das Detail-Panel hindurch zu erkennen ist. Dies hat nicht nur den Sinn, dass der Benutzer weiterhin in der Lage ist, aus dem Scatterplot Informationen abzulesen, sondern es geht vor allem darum, dass klar kommuniziert wird, dass der Anwender durch das Öffnen eines Detail-Panels lediglich die Repräsentation eines einzigen Datenobjekts verändert hat und der zu Grunde liegende Scatterplot unverändert weiterbesteht. Gerade bei der höchsten Zoom-Stufe in den Detailansichten, vor allem auch bei Multiple Datapoints, liegen, wie später noch ausgeführt wird, mehrere Panels übereinander und somit auch über dem Scatterplot. In diesem Fall kann der Benutzer kaum noch etwas von den Datenpunkten im Scatterplot erkennen. Trotzdem wird dem Benutzer, durch das Animieren der Zoom-Bewegungen und die halb-transparenten Detail-Panels, das Gefühl vermittelt, in Kontrolle der HyperScatter-Visualisierung zu sein, da er durch das aufgebaute mentale Modell weiss, was angezeigt wird, und was hinter den Detail-Panels zu sehen wäre.

Der HyperScatter ist für die Nutzung an einem Desktop-Computer konzipiert. Es wird darum für die Benutzerinteraktion von einer normalen Computertastatur und einer Maus mit mindestens zwei Tasten ausgegangen. Wie auch bei der HyperGrid, werden im HyperScatter Zoom-Bewegungen durch Klicken mit der Maustaste ausgelöst. Dabei wird durch einen Klick mit der linken Maustaste die Detailansicht aufgezoomt, also mehr Detailinformationen angezeigt, durch Klicken mit der rechten Maustaste wird entsprechend aus der Detailansicht hinausgezoomt (siehe Abbildung 28). Im Unterschied zur HyperGrid gibt es im HyperScatter aber vorgegebene Zoomstufen und das Zooming wird durch Klicken ausgelöst. Bei der HyperGrid ist es so, dass eine HyperGrid-Zelle so lange vergrößert bzw. verkleinert wird, bis der Anwender die jeweilige Maustaste wieder loslässt.



Abbildung 28: In MedioVis wird durch einen Linksklick ein Zoom-In- und durch einen Rechtsklick eine Zoom-Out-Aktion ausgeführt.

Um die Interaktion mit dem HyperScatter noch etwas zu beschleunigen, wird das Zoomen in die erste Detailstufe bereits durch den Mouseover-Event ausgelöst, wenn der Benutzer mit der Maus über einen Datenpunkt fährt (ähnlich wie bei einem Tooltip). Dieses effiziente Detail-On-Demand-Konzept unterstützt ausserdem den Browsing-Ansatz beim

Explorieren der Treffermenge, da die Hemmschwelle zur Erkundung der Detailinformationen von Datenobjekten niedrig gehalten wird. So kann ein Benutzer durch die leicht zugänglichen Detailinformationen für die weitere Analyse des Datenraumes inspiriert werden.

5.3.3.3 Detail-Zoom von Single Datapoints

Abbildung 29 zeigt die drei Zoomstufen beim Hineinzoomen in die Detailansicht des Filmes *Full Metal Jacket*. Im Fall der Trefferdarstellung, wie sie in der Abbildung zu sehen ist, wurde in MedioVis nach *Stanley Kubrick* gesucht, welcher unter anderem auch der Regisseur dieses Filmes war. In der obersten Abbildung ist der Benutzer mit der Maus über das Medientyp-Symbol des Filmes gefahren, worauf das Detail-Panel zum Film in der ersten Zoomstufe aufgezoomt wurde. Darauf sind das Filmposter und der Filmtitel zu sehen. Diese beiden Attribute werden als die markantesten Dimensionen bei der Analyse von Datenobjekten aus MedioVis angesehen und deshalb schon auf der ersten Zoomstufe angezeigt. Zu diesem Zeitpunkt werden also schon fünf Attribute des Filmes visualisiert: Titel, Filmposter, Medientyp (durch das Medientyp-Symbol: Im Beispiel handelt es sich um eine DVD) und die beiden auf die Achsen des HyperScatters gemapten Attribute (in diesem Fall das Erscheinungsjahr und die Sprache des Filmes). Zusätzlich dazu ist in der linken oberen Ecke des Detail-Panels eine Checkbox zu sehen, mit der das Datenobjekt selektiert werden kann. Darauf wird im Abschnitt 5.3.6 eingegangen.

Falls der Benutzer an weiteren Detailinformationen zu diesem Film interessiert ist, klickt er mit der linken Maustaste auf das Detail-Panel und die Ansicht zoomt weiter auf, bis zur zweiten Zoomstufe (siehe Abbildung 29, Mitte). Hier werden bereits alle Attribute des Datenobjekts dargestellt, wobei für einige nicht genügend Platz zur Verfügung steht, um diese vollständig anzuzeigen. Auch hier ist links oben wieder die Checkbox zu erkennen, mit der dieses Datenobjekt selektiert werden kann. Die übrigen Attribute sind analog zur HyperGrid in drei Spalten aufgeteilt. Auch die Verteilung der Attribute innerhalb der Spalten ist analog zur HyperGrid gehalten. Der Grund dafür ist, dass in MedioVis später beide Visualisierungen, HyperScatter und HyperGrid, angeboten werden sollen. Der Benutzer soll selber entscheiden, welche der beiden Visualisierungen er für die momentane Aufgabe verwenden will. Um ihm das Wechseln zwischen den beiden Visualisierungen zu vereinfachen, werden die beiden Visualisierungen nach ganz ähnlichen Konzepten aufgebaut. Dies äussert sich zum Beispiel in der Umsetzung der Zoom-Interaktion mittels linker

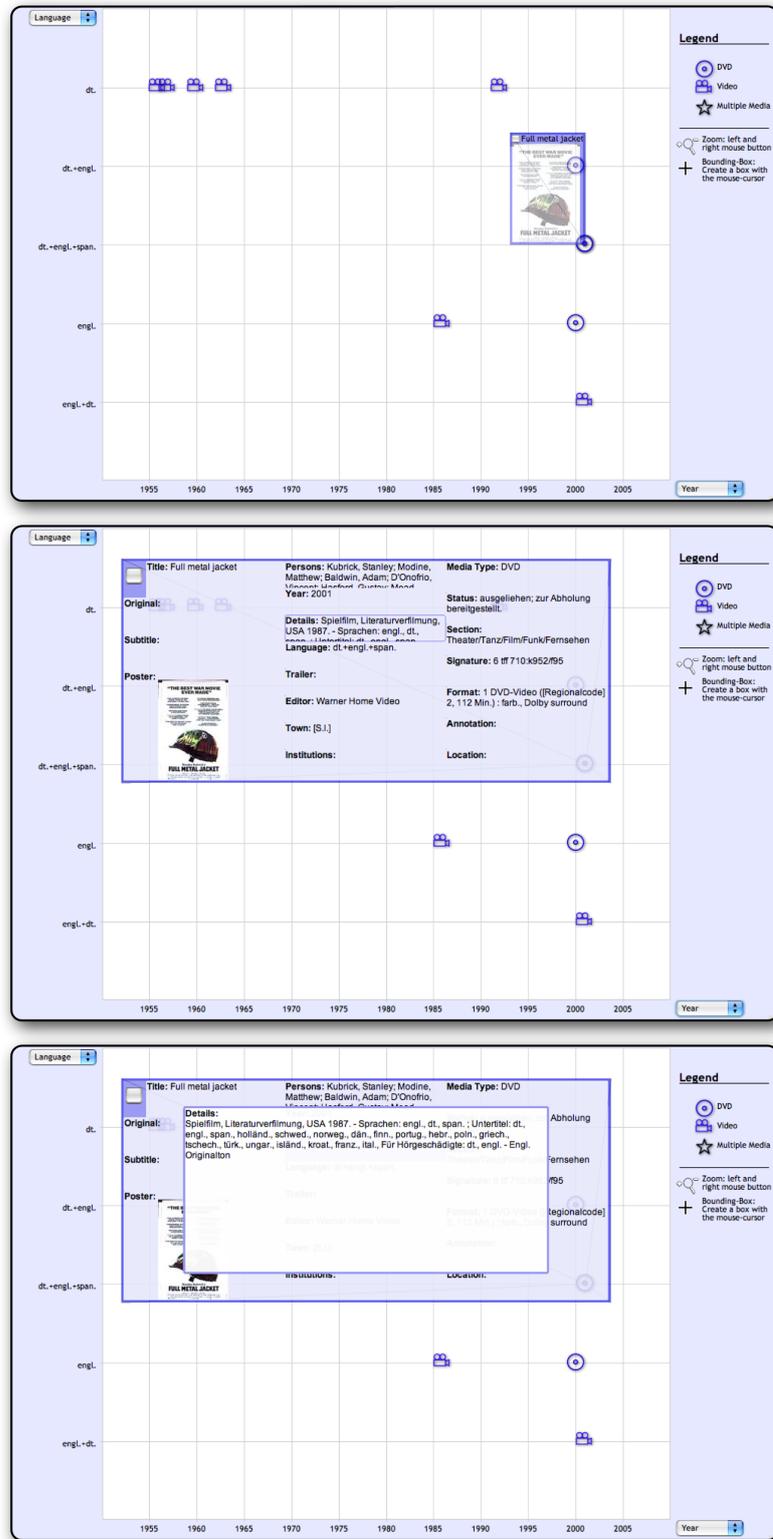


Abbildung 29: Der Detail-Zoom eines einzelnen Datenobjekt im HyperScatter, aufgeteilt in drei Zoomstufen.

und rechter Maustaste oder eben hier durch das analoge Anordnen der einzelnen Attribute der Datenobjekte in der Detailansicht.

Bei dem hier verwendeten Mediotheksszenario haben die einzelnen Datenobjekte eine recht überschaubare Anzahl an Attributen. Trotzdem kommt es oft vor, dass zur Darstellung von gewissen Attributen in der Detailansicht nicht genügend Platz zur Verfügung steht. Im HyperScatter ist daher jedes Attribut in der Detailansicht wieder auf einem eigenen Panel verortet. Wenn ein Attribut zu viel Text beinhaltet, um in der normalen Detailansicht (zweite Zoomstufe) vollständig angezeigt werden zu können, kann der Benutzer durch einen weiteren Linksklick auf das Attribut-Panel dieses einzelnen Attribut herausgreifen, um es genauer zu betrachten. Das Panel wird in die Mitte der Detailansicht verschoben und um eine weitere Stufe vergrößert (siehe Abbildung 29, unten). Im Beispiel wurde das Attribut *Details* weiter aufgezoomt. Durch einen Linksklick auf eines der anderen Attribut-Panel, welche immer noch am Rand der Detailansicht zu sehen sind, wird das aktuell aufgezoomte Detail-Panel wieder in seine ursprüngliche Position und Grösse zurückgezoomt und zugleich nimmt das angeklickte Panel den Platz in der Mitte ein. So können die meisten Attribute mit nur einem Klick nacheinander aufgezoomt werden. Nur bei den Attributen, die in der mittleren Spalte und auf mittlerer Höhe angeordnet sind, muss das aktuell aufgezoomte Attribut zuerst durch einen Rechtsklick auf dessen Panel zurückgezoomt werden, bevor das darunterliegende Panel herangezoomt werden kann. Durch einen Rechtsklick auf die Detailansicht wird aus dieser schliesslich wieder herausgezoomt und der Benutzer kann das nächste Datenobjekt genauer unter die Lupe nehmen.

Bei allen beschriebenen Zoom-Animationen wird ein organisches Zooming verwendet. Dabei wird eine variable Zoomgeschwindigkeit mit einer konstanten Animationszeit kombiniert. Die Animationen dauern jeweils ca. eine halbe Sekunde und werden dabei am Anfang beschleunigt und am Ende wieder abgebremst. Durch ihre kurze Dauer fallen sie dem Benutzer nicht störend auf, weil für ihn keine Wartezeiten auftreten. Dadurch, dass die erste Zoomstufe bereits durch das Überfahren eines Datenpunktes mit der Maus aufgezoomt wird, ist ausserdem nur ein einziger Mausklick nötig, um in die zweite Zoomstufe der Detailansicht zu gelangen, in welcher bereits alle Attribute eines Datenobjekts angezeigt werden.

5.3.3.4 Detail-Zoom von Multiple Datapoints

Bei Multiple Datapoints besteht im HyperScatter das Problem, dass die zugehörigen Datenpunkte nicht mit der Maus überfahren werden können, da sie einander überdecken. Folglich sind auch ihre Detailinformationen nicht über das normale semantische Zooming zugänglich. Deshalb wird im HyperScatter bei Multiple Datapoints eine zusätzliche Zoomstufe eingeführt, welche in Abbildung 30 dargestellt ist. Beim Überfahren eines Multiple Datapoints (im HyperScatter durch einen Stern repräsentiert) mit der Maus, wird ein Zoom-Panel geöffnet, in dem Angaben über die Datenpunkte angezeigt werden, welche von dem Multiple Datapoint zusammengefasst werden. Dabei handelt es sich um die Anzahl an Datenobjekten jedes Medientyps, der in diesem Multiple Datapoint enthalten ist. In Abbildung 30 oben, sind dies beispielsweise acht Videokassetten und zwei DVDs.

Ebenfalls durch Linksklicken auf das Panel wird dieses weiter aufgezoomt und zeigt jetzt eine genauere Übersicht über die Datenobjekte des Multiple Datapoints (siehe Abbildung 30, Mitte). Das Panel ist jetzt einige Pixel breiter und höher als ein Detail-Panel eines normalen Single Datapoints. Dies bietet Platz für zehn Single-Datapoint-Detail-Panels in der ersten Zoomstufe, wie in der Abbildung zu erkennen ist. Diese Detail-Panels weisen das identische Interaktionskonzept auf wie die Detail-Panels von normalen Single Datapoints. Jedes dieser Datenobjekte kann durch einen Linksklick in die zweite Zoomstufe gebracht werden. Dies ist in Abbildung 30, unten zu sehen. Von da aus, kann jedes der Attribut-Panel noch einmal vergrössert werden. Im Wesentlichen wird im HyperScatter bei Multiple Datapoints also einfach eine Zoomstufe vorgeschaltet.

Wenn, wie in Abbildung 30 unten, ein Datenobjekt gross in der Mitte dargestellt wird, kann durch einen Linksklick auf eines der restlichen neun Datenobjekte, welche am Rand teilweise zu sehen sind, das aktuell herangezoomte Datenobjekt wieder verkleinert und dafür gleichzeitig das angeklickte Datenobjekt ins Zentrum geholt werden.

Um innerhalb der Visualisierung die Konsistenz und das Verständnis der Konzepte beim Benutzer zu erhöhen, werden die kleinen Detail-Panel der Single Datapoints innerhalb eines Multiple Datapoint (Abbildung 30, Mitte) genau gleich gross gezeichnet wie bei einem normalen Single Datapoint (Abbildung 29, oben). Dadurch ist es aber nur möglich zehn solcher Detail-Panels in die zweite Zoomstufe eines Multiple Datapoints zu integrieren. Da es aber vorkommen kann, dass weit mehr Datenobjekt in einem Multiple Datapoint

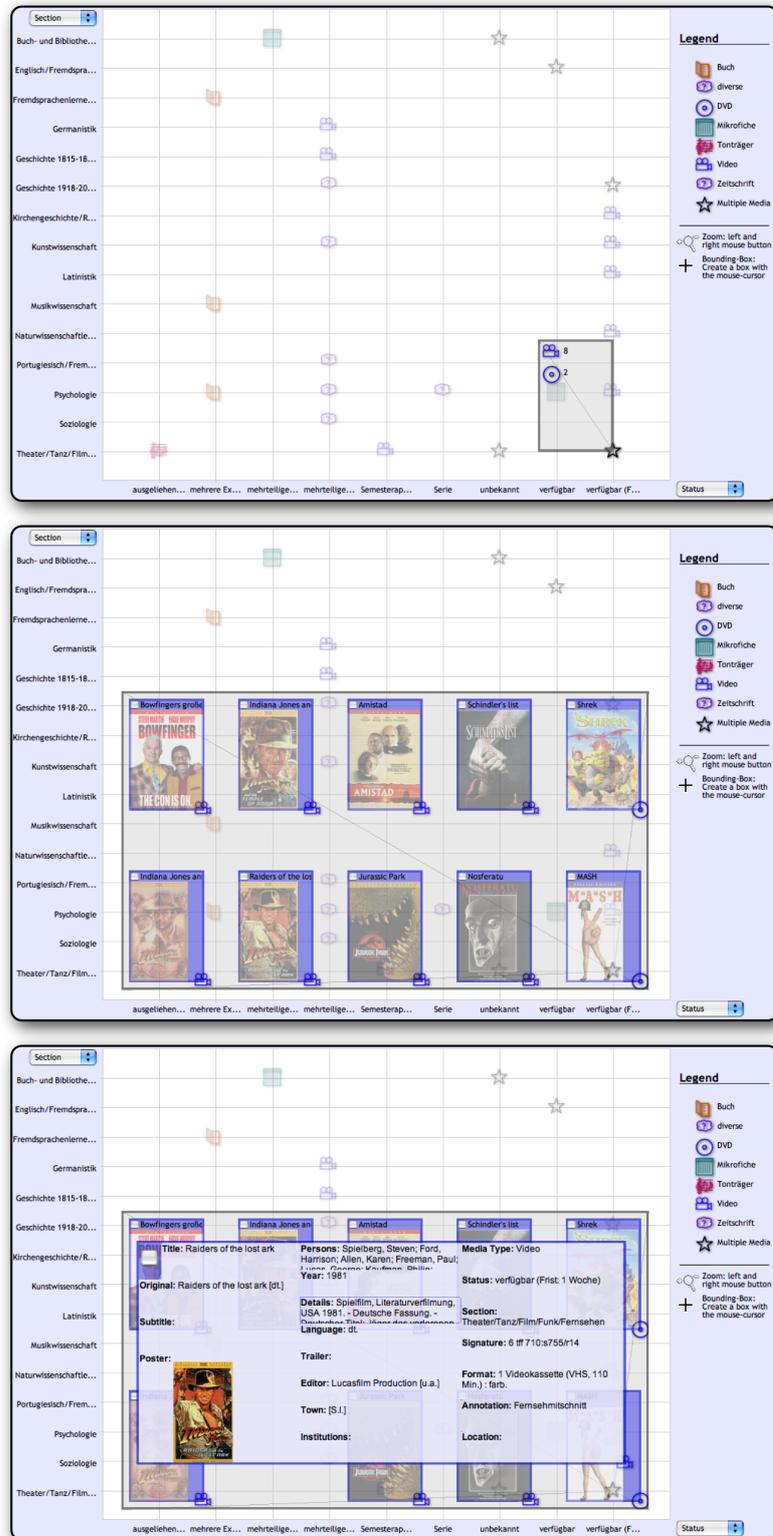


Abbildung 30: Die Darstellung eines Multiple Datapoints in den verschiedenen Zoomstufen.

zusammengefasst werden müssen, wird, wie in Abbildung 31 zu sehen ist, eine Art Blättern durch mehrere Seiten von Datenobjekten innerhalb des Multiple Datapoint angeboten.

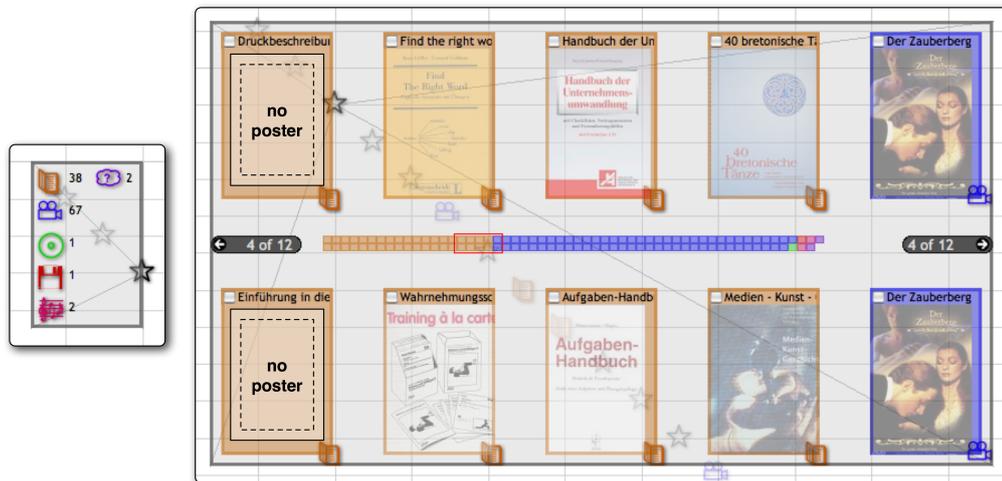


Abbildung 31: Die Repräsentation eines Multiple Datapoints im HyperScatter.

In dem Multiple Datapoint der Abbildung 31 befinden sich 111 Datenobjekte (38 Bücher, 67 Videokassetten, eine CD, eine Diskette, zwei Tonträger und zwei Datenobjekte unbekannt Typs). Das alles ist aus der ersten Zoomstufe des Multiple Datapoint herauszulesen, welche beim Überfahren des Punktes im HyperScatter angezeigt wird. Wenn der Benutzer diesen Multiple Datapoint aufzoomt, erhält er ein Panel, auf dem ihm die ersten zehn Datenobjekte, entsprechend der vorherigen Ausführungen, dargestellt werden. Da in diesem Multiple Datapoint aber mehr als zehn Datenobjekt enthalten sind, wird dem Benutzer zusätzlich durch zwei Buttons links und rechts des Multiple-Datapoint-Panels die Möglichkeit geboten, sich die nächsten, bzw. die vorherigen zehn Datenobjekte im Panel anzeigen zu lassen. Auf diesen Buttons wird neben einem Pfeil, welcher die Richtung des Blätterns anzeigt, auch die aktuelle Position innerhalb des Multiple Datapoints angezeigt. Im Beispiel befindet sich der Benutzer gerade auf der vierten von zwölf Seiten.

Zwischen diesen beiden Buttons ist eine zusätzliche Übersicht über die Datenobjekte, welche von dem Multiple Datapoint aggregiert werden, angebracht (siehe Abbildung 31, rechts). Darauf wird jedes Datenobjekt durch ein kleines Rechteck in der Farbe seines Medientyp-Symbols repräsentiert. Diese Rechtecke werden dabei genau so angeordnet, wie die Datenobjekte im Panel zu liegen kommen würden, wenn dieses statt nur zwei auf fünf Datenobjekte, eine unendlich lange Kette von Datenobjekten (immer zwei übereinander) anzeigen könnte. Auf dieser Kette von Rechtecken wird durch einen roten Rahmen der

aktuelle Ausschnitt, der im Panel angezeigt wird, markiert. Betrachtet man im Beispiel (Abbildung 31, rechts) diesen Rahmen, stellt man also fest, dass von links nach rechts zuerst acht braune Rechtecke (also Bücher) dann zwei blaue Rechtecke (also Videokassetten) im Panel dargestellt werden müssten, was offensichtlich auch der Fall ist. Die einzelnen Datenobjekte sind dabei innerhalb dieser Darstellung nach dem Medientyp geordnet. So muss ein Benutzer, der sich zum Beispiel nur für Filme interessiert nicht durch alle Datenobjekte durchblättern, sondern kann nur die Stelle anschauen, an denen die Videokassetten und die DVDs angezeigt werden.

Diese Art des Betrachtens der Datenobjekte eines Multiple Datapoints mag im ersten Moment etwas mühsam wirken. Die Exploration solcher listenähnlichen Trefferdarstellungen war ja ein Kritikpunkt an herkömmlichen Suchsystemen. In diesem Fall muss man aber berücksichtigen, dass alle Datenpunkte, die zu einem bestimmten Multiple Datapoint gehören, im HyperScatter tatsächlich genau auf derselben Stelle zu liegen gekommen sind. Das heisst, dass diese auch wirklich alle die gleiche Relevanz bezüglich der auf die beiden HyperScatter-Achsen abgetragenen Attribute aufweisen. Eine weitere komplexe Aufbereitung dieser Datenpunkte, zum Beispiel anhand einer weiteren Scatterplot-Visualisierung mit anderen Achsenbelegungen, würde die HyperScatter-Visualisierung komplizierter machen als beabsichtigt. Deshalb wird im HyperScatter-Konzept das lineare Durchblättern durch eine „Liste“, wie es oben beschrieben wird, als die am besten geeignete Vorgehensweise zur Darstellung der Datenobjekte eines Multiple Datapoint angesehen.

5.3.3.5 Browser-Komponente im HyperScatter

In Kapitel 4.2.3 wurde mit dem HyperGrid-Browser eine Komponente vorgestellt, die es dem Benutzer ermöglicht, eine browsing-orientierte Analyse und Suche auf dem Datenbestand der Mediothek durchzuführen. Dabei können über den Browser auch externe Datenquellen in den MedioVis-Client integriert werden. Im HyperScatter soll analog zur HyperGrid ebenfalls eine Browser-Komponente in die Visualisierung integriert werden, welche die gleiche Aufgabe wie der HyperGrid-Browser übernimmt. Die Entwicklung dieser Browser-Komponente ist aber noch nicht so weit fortgeschritten, dass im Rahmen dieser Arbeit schon darüber berichtet werden kann.

5.3.3.6 Technische Umsetzung des animierten Zoomings von Java-Components im HyperScatter

In einem weiteren kurzen Exkurs in die technische Ebene der Umsetzung des HyperScatters soll vorgestellt werden, wie die Animationen der verschiedenen Detail-Panels beim semantischen Zooming implementiert worden sind. Im HyperScatter sind alle zoombaren Detail-Panels (Single-Datapoint-Detail-Panel, Attribut-Panel und Multiple-Datapoint-Detail-Panel) von der Java-Klasse `java.awt.Component`²¹ abgeleitet, welche die Methode `setBounds(int x, int y, int width, int height)` bereitstellt. Dadurch kann das semantische Zooming im HyperScatter durch eine einzige Klasse, die Animator-Klasse, zentral implementiert werden. Dies ist schematisch in Abbildung 32 dargestellt.

Die Animator-Klasse übernimmt neben dem eigentlichen Zooming der Detail-Panels auch die Koordination von verschiedenen Zoom-Animationen. Im HyperScatter ist es zum Beispiel nicht möglich, dass ein Benutzer von zwei Filmen gleichzeitig die Detailansicht aufzoomt, da dafür nicht genügend Platz auf dem Bildschirm vorhanden ist und so Überdeckungen der Detailansichten die Folge wären. Es kann also immer nur ein Detail-Panel auf einmal geöffnet werden. Da die Animationen der Zoom-Vorgänge jeweils in eigenen Threads ablaufen, muss beim Starten eines Detail-Zooms kontrolliert werden, ob nicht schon eine andere Zoom-Animation am ablaufen ist. Ist dies der Fall, muss diese abgebrochen werden, bevor in die Detailansicht des zuletzt angeklickten Filmes hineingezoomt wird.

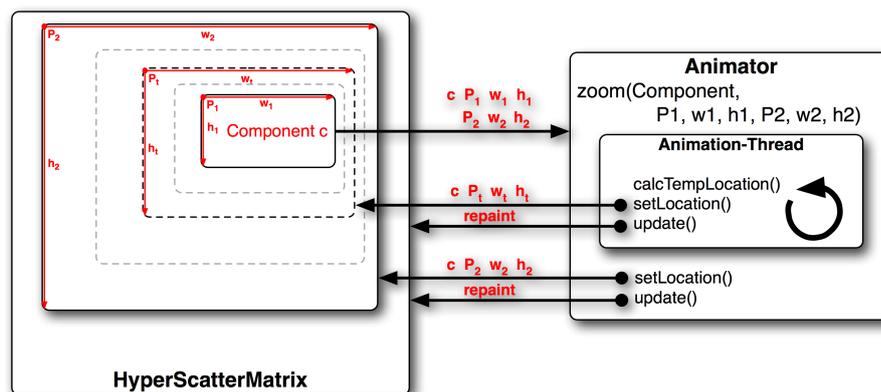


Abbildung 32: Im HyperScatter können alle Java-Components, die auf dem Scatterplot-Panel (HyperScatterMatrix) gezoomt werden, einfach mit der Animator-Klasse einheitlich animiert gezoomt werden.

²¹<http://java.sun.com/j2se/1.5.0/docs/api/java/awt/Component.html>

Sobald dieser Vorgang erledigt ist, wird dem Animator aus der Visualisierung das zu zoomende Detail-Panel (Component) (ganz egal um was für eines es sich dabei handelt), die Start- und die Endposition (P1 und P2) und die Start- und Endgrösse (w1, h1, w2, h2 — mit w für Breite und h für Höhe) des Panels in der Methode `zoom(Component, P1, w1, h1, P2, w2, h2)` übergeben. In dieser Methode wird dann in einem eigenen Thread jeweils die nächste temporäre Position und Grösse des animierten Panels berechnet, über die `setBounds()`-Methode des Panels an die Visualisierung übertragen und ein `repaint()` aufgerufen. Die temporären Positionen und Grössen des zoomenden Panels berechnen sich aus den Start- und Endwerten von Position und Grösse und einem Zahlen-Array, durch das die Beschleunigung und das Abbremsen der Zoom-Animation definiert wird. Am Anfang ändern sich Position und Grösse von Animationsschritt zu Animationsschritt nur wenig. Die Änderungen werden gegen Mitte der Animation immer grösser und nehmen dann gegen Ende wieder ab. Im letzten Schritt, wird der Thread beendet, die Endposition und die Endgrösse des Panels an die Visualisierung übertragen und ein letztes `repaint()` der Visualisierung aufgerufen.

Dies ist in Abbildung 32 schematisch und vereinfacht dargestellt. Links ist der HyperScatter dargestellt, auf dem das Detail-Panel (Component c) liegt. Dieses wird von Position P1 und von der Grösse w1 und h1 auf die Position P2 und die Grösse w2 und h2 gezoomt. Rechts ist die Animator-Klasse zu sehen in welcher die `zoom()`-Methode aufgerufen wird.

5.3.4 Zoom

Mit der Kernaufgabe *Zoom* meint Shneiderman das Konzept, dass ein Benutzer, ausgehend von einer Übersichtsdarstellung einer Kollektion von Datenobjekten, durch Hineinzoomen seine Sicht auf einen Teilbereich der Kollektion einschränken kann um diesen Bereich durch den gewonnenen Bildschirmplatz differenzierter darstellen zu können. Bei MedioVis wird diese Kollektion von Datenobjekten wiederum durch die Treffermenge einer Suchanfrage verkörpert.

Das Betrachten eines Teilbereichs der Treffermenge macht zum Beispiel dann Sinn, wenn ein Benutzer an der genaueren Betrachtung eines Datapoint Clusters, wie es im Kapitel 2 beschrieben wird, interessiert ist, weil er unter den beteiligten Datenobjekten dieses Clu-

sters für ihn relevante Datenobjekte vermutet, diese aber auf Grund von Überdeckungen nicht genau identifizieren kann.

In diesem Fall kann durch das geometrische Hineinzoomen in diesen Teilbereich des HyperScatters der Platz zur Darstellung des Datapoint Clusters vergrößert werden. Dies geschieht allerdings auf Kosten der umliegenden Bereiche des Scatterplots. Es wird bewusst in Kauf genommen, dass durch das geometrische Zooming im HyperScatter die Komplett-Übersicht über die ganze Treffermenge geopfert wird. Anders als beim Detail-Zoom, bringt das geometrische Zooming hier einen Vorteil, denn so können Datapoint Cluster aufgelöst und auch diese Datenpunkte für das semantische Zooming zugänglich gemacht werden.

5.3.4.1 Geometrisches Zooming unter Verwendung einer Bounding-Box

Umgesetzt wird das geometrische Zooming im HyperScatter durch den Einsatz einer Bounding-Box. Diese wird, wie aus anderen Anwendungen gewöhnt, durch das Drücken und Halten der Maustaste und das Loslassen der Taste an einer anderen Stelle auf dem HyperScatter definiert, wie dies in Abbildung 33 in den oberen vier Bildern veranschaulicht wird.

Während des Aufziehens wird innerhalb der Bounding-Box die Anzahl beinhalteneter Datenpunkte angezeigt. Diese Zahl wird ständig aktualisiert, so dass der Benutzer zu jeder Zeit weiss, wie viele Datenobjekte sich gerade innerhalb des Rahmens, der durch die Box aufgespannt wird, befinden. Damit noch klarer wird, in welchen Bereich des HyperScatter gleich hineingezoomt wird, wird das Äussere der Box abgedunkelt.

Sobald der Benutzer die linke Maustaste loslässt, ist die Bounding-Box fertig definiert und es erscheinen zwei Buttons auf der Box, *Select* und *Zoom*. Mittels der Bounding-Box ist es im HyperScatter nicht nur möglich in einen Teilbereich des Scatterplots hineinzuzoomen, sondern es können auf diese Art auch mehrere Datenobjekte selektiert werden. Auf diese Funktionalität wird später im Abschnitt 5.3.6 eingegangen. Um das geometrische Zooming auszulösen, klickt der Benutzer jetzt auf den *Zoom*-Button. Auch dieses Zooming wird durch eine Animation verständlich gemacht. Dabei werden die Bereiche des HyperScatters, welche ausserhalb der Bounding-Box gelegen haben mitsamt den daraufliegenden Datenpunkten, über die Visualisierungsränder, aus dem Bild geschoben.

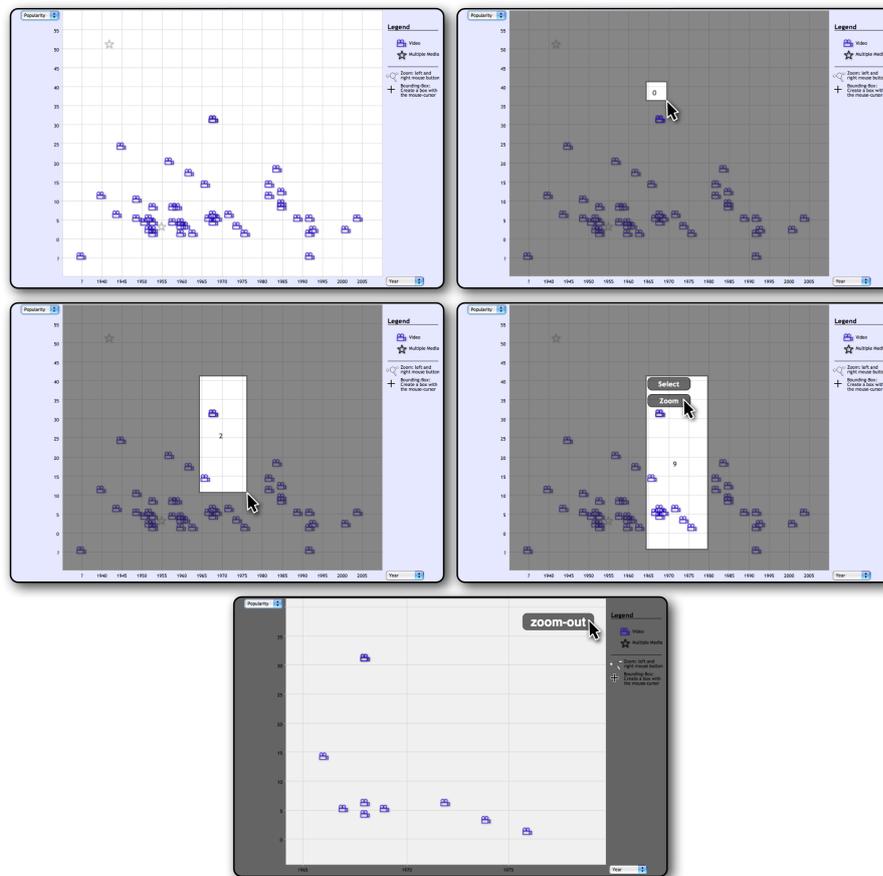


Abbildung 33: Durch ziehen einer Bounding-Box (obere vier Bilder) kann im HyperScatter in diesen Bereich geometrisch hineingezoomt werden (unteres Bild).

Sobald die Animation fertig ist, wird nur noch der vorher durch die Bounding-Box definierte Teilbereich des HyperScatter auf der vollen Bildschirmfläche angezeigt (siehe Abbildung 33, unten). Wie man in der Abbildung erkennen kann, liegen nun die Datenpunkte weiter auseinander. Falls an dieser Stelle immer noch Datapoint Cluster vorhanden wären, könnte der Benutzer jetzt eine neue Bounding-Box definieren und so eine erneute Zoom-Aktion auslösen. Falls der Benutzer hingegen wieder zurück in die Gesamtübersicht der Treffermenge gelangen will, klickt er mit der Maustaste auf den Zoom-Out-Button, welcher oben rechts auf der Scatterplot-Fläche erscheint, sobald in den HyperScatter hineingezoomt wurde.

Die Höhe und Breite der Bounding-Box kann vom Benutzer frei gewählt werden. Durch das Auslösen des Zoomings, wird der Bereich der Box automatisch so gedehnt, dass er am

Schluss den gesamten Bildschirmplatz ausfüllt. Dadurch ist es auch möglich nur in einer Dimension des HyperScatter zu zoomen, indem die Bounding-Box entsprechend definiert wird.

Beim Ziehen einer Bounding-Box wird der Bereich ausserhalb der Box abgedunkelt. Analog dazu werden nach dem Ausführen eines geometrischen Zooms im HyperScatters die Achsen- und der Legenden-Hintergrund ebenfalls verdunkelt (siehe Abbildung 33, unten). Dies soll verdeutlichen, dass nur ein Teilbereich des Scatterplots sichtbar ist. Ein alternativer Ansatz wäre das Verzerren des HyperScatters im Sinne einer Fisheye-Distortion [Fur86], wie beispielsweise beim ZuiScat getestet wurde [BGR06]. Hier würde dann der gezoomte Bereich des HyperScatters im Zentrum dargestellt, der Bereich der zuvor ausserhalb der Bounding-Box gelegen hat, würde aber verzerrt am Rand, in einem Kontext-Bereich, weiterhin dargestellt. Diese Möglichkeit sollte bei der Weiterentwicklung des HyperScatter in Betracht gezogen werden. Im Rahmen dieser Arbeit wurde dieses Konzept jedoch nicht umgesetzt.

5.3.5 Filter

Das *Filtern* ist ein Konzept, das den Benutzer in die Lage versetzt, im Sinne von Dynamic Queries in einer Kollektion von Datenobjekten schnell diejenigen Objekte weg zu filtern, die für ihn uninteressant sind, so dass vorwiegend (im Idealfall nur noch) relevante Daten übrig bleiben. Bei der Anwendung auf MedioVis würde das bedeuten, dass der Anwender nach einer Suche mit Hilfe von Filtern die irrelevanten Datenobjekte, welche ebenfalls auf die Suchanfrage angesprochen haben, aus der Treffermenge herausfiltern kann, in dem er zum Beispiel nachträglich durch einen Range-Slider das Erscheinungsjahr auf die für ihn relevante Zeitspanne einschränkt.

Filter-Widgets können verschiedene Formen haben, wie dies zum Beispiel von Ahlberg und Shneiderman anhand des FilmFinders gezeigt wird [AS94a]. Von einfachen Textfeldern über Slider, Range-Slider, Checkboxes oder visuellen Tools wie zum Beispiel Bounding-Boxes, gibt es viele verschiedene Möglichkeiten, eine Filter-Funktionalität zu implementieren. Dabei ist es wichtig, dass die visuelle Darstellung der Datenkollektion dynamisch auf Benutzerinteraktionen mit den Filter-Tools reagiert und der Bildschirm aktualisiert wird. Datenobjekte, die auf ein negatives Filterkriterium ansprechen, werden also sofort ausgeblendet. Der Vorteil von Filtern (im Sinne von Dynamic Queries) gegenüber ei-

ner erneuten, verfeinerten Suchanfrage ist, dass sie die Interaktivität einer Anwendung fördern, indem keine zeitaufwändige Datenbank-Anfragen gestellt werden müssen. Durch das Filtern werden lediglich nachträglich Datenobjekte ausgeblendet, die bereits im lokalen Arbeitsspeicher der Client-Anwendung gespeichert sind.

Eine besondere Möglichkeit Dynamic Queries zu nutzen ist es, in einem Suchsystem überhaupt keine eigentliche textuelle Suchanfrage durchführen zu müssen, sondern, ausgehend vom gesamten Datenbestand, direkt mit Dynamic Queries zu arbeiten. Das heisst, dass zu Beginn der Suche im Prinzip der gesamte Datenraum angezeigt wird, und erst nach und nach durch das Hinzufügen von Filtern die Treffermenge eingeschränkt wird, bis nur noch relevante Datenobjekte übrig sind. Dieses Konzept wird unter anderem in vielen Desktop-Suchsystemen wie der Yahoo! Desktop Search²² oder Spotlight²³, der in Mac OSX integrierten Desktop-Suche angewendet. Bei Web und Katalogsuchsystemen besteht zur Zeit noch das Problem, dass durch die relativ langsame Datenanbindung, meistens über das Internet, die Antwortzeiten zu langsam sind, als dass eine wirklich direkt-manipulative Interaktion mit dem System möglich wäre. Der Benutzer soll bei der Formulierung von Dynamic Queries keinen Wartezeiten ausgesetzt werden.

5.3.5.1 Globales Filtersystem in MedioVis

Für die HyperGrid wurde im Kapitel 4 der Spaltenfilter vorgestellt, welcher direkt in die Visualisierung integriert ist. Die Idee dabei ist, dass es dem Benutzer leichter fallen soll den Spaltenfilter zur Einschränkung eines Suchresultates zu benutzen, da die Anwendung des Spaltenfilters, durch die räumliche Nähe zur Resultatdarstellung leichter zu verstehen ist. Diese These konnte allerdings bis jetzt durch die Analyse von Benutzerinteraktionsdaten nicht bestätigt werden.

Ein Problem bei der Integration von Dynamic-Query-Filtern in die jeweiligen Visualisierungen ergibt sich, wenn in MedioVis vom Benutzer zwischen mehreren solchen Visualisierungen umgeschaltet werden kann, so dass diese nicht im Sinne einer Multiple Coordinated View [NS00] parallel angezeigt werden, sondern jeweils eine einzige Visualisierung den gesamten zur Verfügung stehenden Bildschirmplatz in Anspruch nimmt.

²²<http://desktop.yahoo.com/>

²³<http://www.apple.com/macosx/features/spotlight/>

Das angesprochene Problem ergibt sich daraus, dass nicht klar ist, was mit den in einer Visualisierung (beispielsweise in den Spaltenfiltern der HyperGrid) definierten Filterkriterien beim Umschalten auf eine andere Visualisierung (wie dem HyperScatter) passieren soll. Sollen die herausgefilterten Datenobjekte weiterhin unsichtbar bleiben, obwohl der Filter, der dafür verantwortlich ist, gar nicht mehr sichtbar ist. Oder sollen alle Filter zurückgesetzt werden, so dass wieder die komplette Treffermenge sichtbar ist, was auch zu Verwirrung führen kann, da der Benutzer möglicherweise im Glauben ist, dass die zuvor definierten Filterkriterien immer noch aktiv sind.

Bei der Konzeption des HyperScatters wurde aber genau diese Integration der Visualisierung in das MedioVis-Framework gewählt, da der HyperScatter seine Vorteile nur dann ausspielen kann, wenn ihm genug Platz zur Verfügung steht. Wie im Anwendungsszenario herausgestellt wurde, gilt das selbe auch für die HyperGrid.

Darum wird mit der Konzeption des HyperScatter auch vorgeschlagen, in MedioVis ein globales Filterkonzept einzuführen, welches Filter-Widgets von den einzelnen Visualisierungen loslöst und diese MedioVis-weit zugänglich macht, unabhängig davon, welche Visualisierung gerade zur Anzeige der Suchtreffer ausgewählt wurde. Diese Filter würden dann, ähnlich wie die MedioVis-Suchmaske, unabhängig von den Visualisierungen in einem festen Bereich des Anwendungsfensters dargestellt werden.

Im Rahmen dieser Arbeit soll nicht weiter auf diese Thematik eingegangen werden, da ein solches globales Filterkonzept letztlich nicht direkt mit der HyperScatter-Visualisierung zusammenhängt. Trotzdem soll darauf hingewiesen werden, dass die Verwendung von Filtern, um Dynamic Queries auf einer grossen Treffermenge auszuführen, eine wichtige Rolle in einem visuellen Suchsystem wie MedioVis einnehmen soll. Durch das nachträgliche Einschränken der Datenobjekte einer Suchanfrage kann das Auftreten und die Grösse von Multiple Datapoints und von Datapoint Clustern verringert werden, indem schlicht und einfach die Anzahl an dargestellten Datenpunkten im HyperScatter reduziert wird.

5.3.6 Extract

Auch bei der *Extraktion* handelt es sich um einen Service, der nicht auf Visualisierungsebene, sondern auf Ebene des Suchsystems, in diesem Fall also MedioVis, global angeboten werden muss. Das Ziel dabei ist es, dass ein Benutzer, der zum Teil über mehrere Suchan-

fragen hinweg und durch intensive Nutzung der Analysewerkzeuge und Visualisierungen des Suchsystems, eine Kollektion von für ihn relevanten Datenobjekt gesammelt hat, die gewonnen Informationen auch zur weiteren Nutzung extrahieren kann. Dies kann vom einfachen Ausdrucken einer Liste mit Detailinformationen bis zur Übertragung der Informationen in eine weitere Applikation reichen.

Auch ist in MedioVis vorgesehen, dass einzelne Datenobjekte selektiert werden können. Dabei wird der Selektionsstatus jedes einzelnen Objektes nicht in der Visualisierung, sondern in dem Datenobjekt selber gespeichert, so dass alle Visualisierungen die dem Benutzer zur Verfügung stehen, das Objekt dementsprechend selektiert oder nicht-selektiert darstellen können. Dadurch spielt es keine Rolle, in welcher Visualisierung ein Datenobjekt selektiert wurde und dem Benutzer ist es möglich, beliebig zwischen den Visualisierungen hin und her zu wechseln, je nach konkreter Aufgabe die er gerade zu erfüllen hat und der Visualisierung die er dazu einsetzen will.

5.3.6.1 Selektion einzelner Datenobjekte

Wie eine Selektion in der HyperGrid ausgeführt wird, wurde schon im Kapitel 4.2.5 behandelt. Da bei der Konzeption vom HyperScatter eine möglichst starke Konsistenz zur HyperGrid angestrebt wird, ist die Selektion eines einzelnen Datenobjekts im HyperScatter praktisch gleich gelöst wie in der HyperGrid. Auch hier wird beim Betrachten der Detailansicht eine einfache Checkbox zur Selektion des Datenobjekts angeboten (siehe Abbildung 34, oben). Diese Checkbox wird dabei in beiden Zoomstufen der Detailansicht angezeigt. So ist es möglich ein Datenobjekt mit nur einem einzigen Mausklick zu selektieren.

Zum Vergleich ist unten in Abbildung 34 eine ganz aufgezoomte HyperGrid-Zelle zu sehen, welche ebenfalls in der linken oberen Ecke eine Checkbox zur Selektion des Datenobjekts aufweist.

5.3.6.2 Selektion mehrerer Datenobjekte

Aufbauend auf der Übersichtsdarstellung des HyperScatters, welche dem Benutzer das schnelle Identifizieren von relevanten Bereichen auf dem Scatterplot ermöglicht, soll auch das Selektieren ganzer Bereiche (bzw. der darin verorteten Datenobjekte) möglich sein. Eine Möglichkeit das zu erreichen, ist mit einer Bounding-Box einen Bereich des

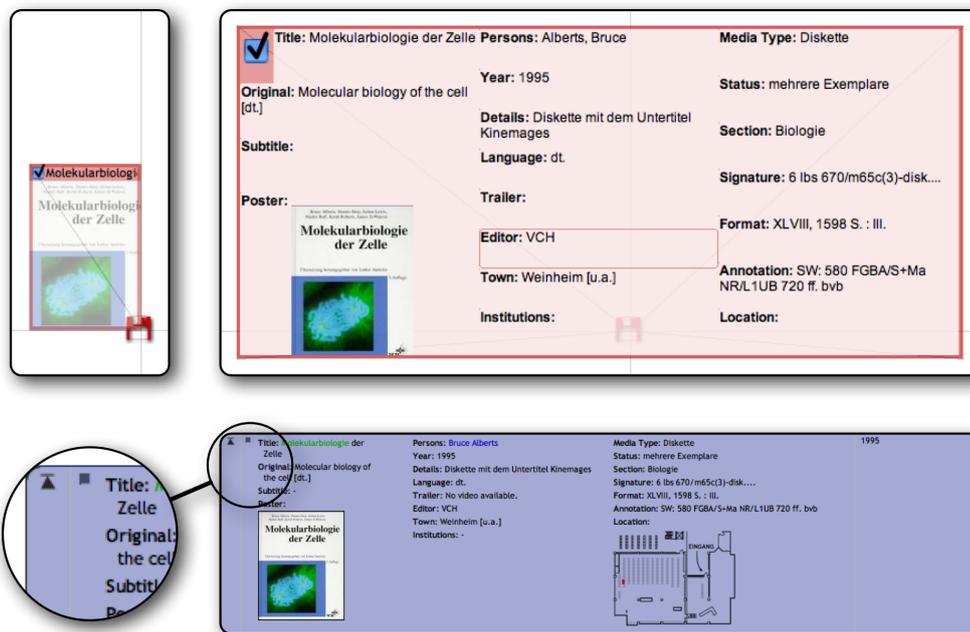


Abbildung 34: Oben: Selektion einzelner Datenobjekte im HyperScatter. Unten: Selektion in der HyperGrid (ebenfalls durch eine Checkbox).

HyperScatters einzurahmen und dann auf den eingerahmten Datenobjekten eine Selektion auszulösen. Zum Beispiel wäre es vorstellbar, dass ein Benutzer aus der Treffermenge zur Suchanfrage *Spielberg* alle DVDs selektieren will, um diese danach als Vorschläge für einen DVD-Abend weiter zu verwenden. Dazu könnte der Anwender in der Trefferdarstellung den Medientyp auf die Y-Achse des HyperScatter legen und dann mit der Bounding-Box alle DVDs einrahmen und selektieren (siehe Abbildung 35).

Da die Bounding-Box schon zur Definition des Zoom-Ausschnittes beim geometrischen Zooming verwendet wird, muss ihre Funktionalität überladen werden. Dies geschieht dadurch, dass sobald die Bounding-Box fertig aufgezo-gen, also die Maustaste losgelassen wird, innerhalb der Box zwei Funktions-Buttons erscheinen (siehe Abbildung 35). Je nach dem, ob der Benutzer nun auf *Select* oder *Zoom* klickt, wird die entsprechende Aktion ausgeführt.

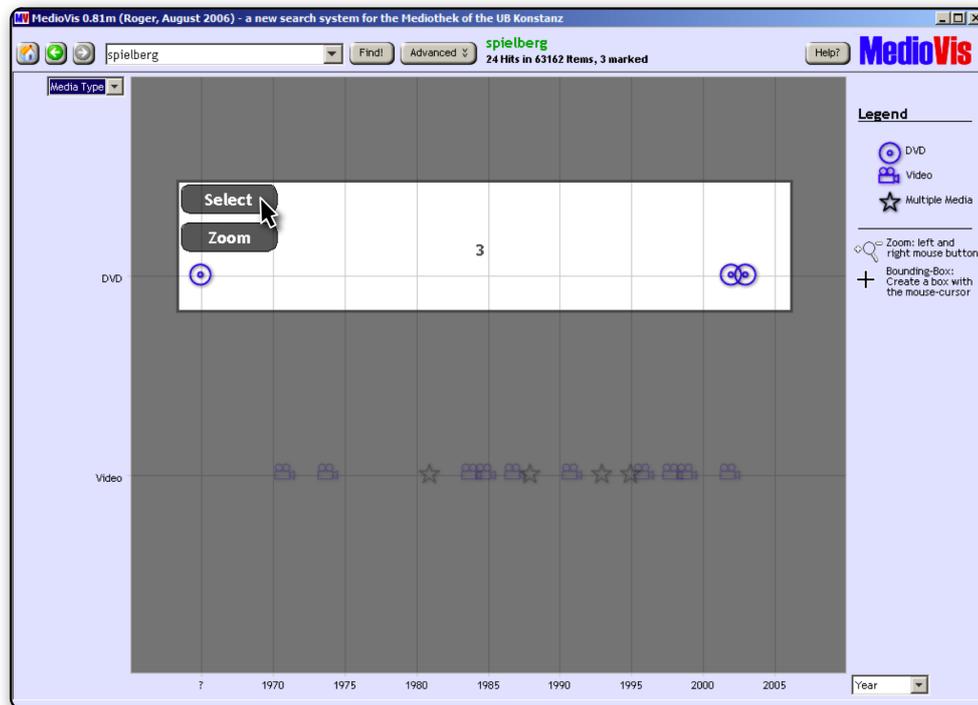


Abbildung 35: Mit der Bounding-Box ist im HyperScatter neben dem geometrischen Zooming auch die Selektion mehrerer Datenobjekt möglich.

5.3.6.3 Weiterverarbeitung von Selektierten Datenobjekten

Wie die selektierten Datenobjekte weiterverarbeitet werden können, wird von MedioVis geregelt. Im Kapitel zur klassischen Benutzung von MedioVis wurde dazu der Warenkorb vorgestellt, in dem einzelne Suchtreffer auch über mehrere Suchen hinweg gespeichert werden können. Bei der Benutzung von MedioVis mit dem HyperScatter soll dieses Konzept weiterverfolgt werden. Es soll also sowohl in der eigentlichen Resultatansicht, welche alle Suchtreffer der letzten Suchanfrage visualisiert, als auch im Warenkorb frei zwischen den verschiedenen in MedioVis zur Verfügung stehenden Visualisierungen, zum Beispiel der HyperGrid und dem HyperScatter, umgeschaltet werden können. Dadurch ist auch eine weitere Exploration und Analyse der Datenobjekte, welche vom Benutzer in den Warenkorb gelegt wurden, möglich.

Ebenfalls soll es weiterhin möglich sein, die Detailinformationen selektierter Datenobjekte auf Papier auszudrucken und sich selbst oder anderen Personen die Informationen per E-

Mail zu zuschicken. Dafür werden die Informationen geeignet aufbereitet und formatiert (siehe Abbildung 36). Die Abbildung zeigt eine Liste mit Informationen zu einigen Filmen der Mediothek. Mit dieser Liste ist es einem Benutzer nun möglich, nacheinander die entsprechenden Filme über die Signatur in den Regalen der Mediothek physisch aufzufinden um sie auszuleihen. Welche Attribute auf so einer Liste angezeigt werden, hängt vom Use-Case ab.

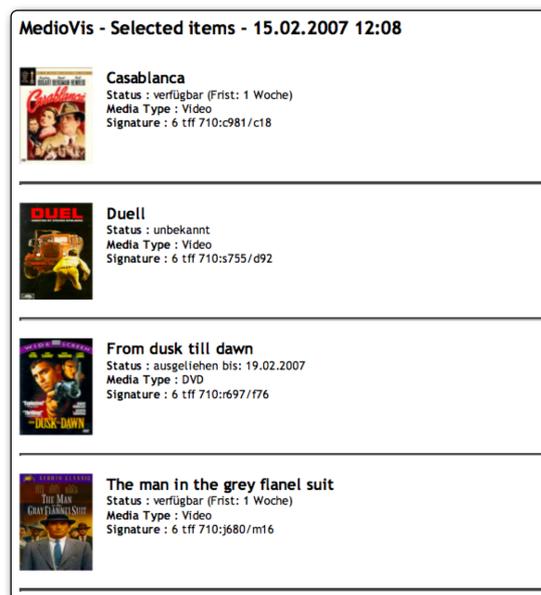


Abbildung 36: Liste der Selektierten Datenobjekt die ausgedruckt werden kann um danach in den Regalen der Mediothek nach diesen Exemplaren zu suchen.

Ein interessanter Gedanke im Zusammenhang mit der Extraktion von Daten aus dem Suchsystem MedioVis wäre in Zukunft auch, dass nicht nur eigentliche Datenobjekten extrahiert werden können, sondern auch Suchanfragen (inklusive allfälliger Filterkriterien). Einem Benutzer wäre es damit jeder Zeit möglich, sich über Neuzugänge der Mediothek in seinen spezifischen Interessengebieten zu informieren, indem er eine gespeicherte Suchanfrage ohne erneutes Formulieren von Schlüsselwörtern und Filterkriterien auswählen und abschicken könnte. Als Resultat würde ihm dann jeweils die aktuelle Treffermenge zur Suchanfrage dargestellt. Dies ist im Use-Case der Mediothek wichtig, da dadurch auch sich schnell ändernde Informationen, wie der Ausleihstatus eines Datenobjekts, mit ihren momentan aktuellen Werten dargestellt werden.

5.3.7 History

Mit dem Konzept der *History* ist es möglich, in einer Anwendung erneut auf Aktionen zu zugreifen, welche von einem Benutzer zu einem früheren Zeitpunkt ausgeführt wurden. Damit können Funktionen wie „Undo“, in Suchsystemen also das Zurückgehen auf bereits getätigte Suchanfragen, oder „Replay“, das erneute Ausführen von Suchanfragen, um eine aktualisierte Treffermenge zu erhalten, angeboten werden. Auch kann der Benutzer durch die Möglichkeit des nachträglichen Verfeinerns von bereits getätigten Suchanfragen bei seiner Arbeit unterstützt werden.

5.3.7.1 Suchanfragen-History

Zur Umsetzung dieses Konzepts, ist in MedioVis eine History-Funktionalität integriert, mit welcher nachträglich wieder auf bereits getätigte Suchanfragen zugegriffen werden kann. Dazu gibt es im oberen Bereich des MedioVis-Clients neben dem Suchfeld zwei Buttons: „Zurück“, gekennzeichnet durch einen Pfeil nach links und „Vorwärts“, gekennzeichnet durch einen Pfeil nach rechts (siehe Abbildung 22, (1)). Möchte ein Benutzer noch einmal die vorherige Suchanfrage bzw. deren Treffermenge betrachten, drückt er einfach auf den Zurück-Button. Um sich in der Such-History wieder nach vorne zu bewegen wird entsprechend der Vorwärts-Button betätigt. Das gleiche Konzept wird bei manchen Web-Browsern (zum Beispiel FireFox²⁴ oder Opera²⁵) dazu verwendet, sich in den besuchten Web-Seiten vor und zurück zu bewegen. Beim Abschicken einer neuen Suchanfrage wird die komplette „Vorwärts-History“ gelöscht und der Benutzer kann dann nur noch entweder zurück gehen oder eine weitere Suchanfrage abschicken.

Ein anderer Ansatz, der in einer weiteren Arbeit entwickelt werden könnte, wäre es, die einzelnen Suchanfragen mit den gleichen Visualisierungen darzustellen, mit denen auch die Suchtreffer dargestellt werden. Damit könnte das Problem des Verlustes der „Vorwärts-History“ gelöst werden. Die Suchanfragen-History würde also ebenfalls mit dem HyperScatter visualisiert werden. Dabei könnte der Zeitpunkt an dem die Anfrage abgeschickt wurde auf die X-Achse und die Grösse der Treffermenge auf die Y-Achse gelegt werden. Ein solches Suchanfragen-History-Konzept wurde beispielsweise im ZuiScat [BR05] umgesetzt.

²⁴<http://www.mozilla.com/en-US/firefox/>

²⁵<http://www.opera.com/>

5.3.7.2 Interaktions-History

Eine weitere Idee, zu der im Rahmen der Entwicklung des HyperScatters einige Versuche unternommen wurden, ist das Einführen einer Interaktions-History. Anders als bei der Suchanfragen-History, sollen hier nicht nur die verschiedenen Suchanfragen und die entsprechenden Treffermengen gespeichert werden, sondern es sollen jegliche Interaktionen, die ein Benutzer mit dem System und den einzelnen Datenobjekten ausführt, aufgezeichnet werden. Das heisst, jede Selektion, jedes Zoomen in eine Detailansicht, jedes geometrische Zoomen auf dem Scatterplot und auch das Ausdrucken und Versenden von E-Mails wird aufgezeichnet.

Jede dieser Aktionen drückt auf die eine oder andere Weise das Interesse eines Benutzers an bestimmten Datenobjekten aus. Das Ausdrucken von Detailinformationen bzw. deren Versenden per E-Mail bedeutet, dass beim Benutzer ein grosses Interesse an diesen Datenobjekten besteht. Aber auch durch das Selektieren und das Hineinzoomen in die Detailinformationen bekundet ein Anwender ein gewisses Interesse an einem Objekt. Wenn hingegen ein Datenobjekt in einer Visualisierung unangetastet bleibt, ist das Interesse des Benutzers bezüglich dieses Objektes als eher gering einzustufen. Durch die verschiedenen Interaktionen wird also ein verschieden grosses Interesse an den Datenobjekten ausgedrückt, was auch als Zahlenwert gefasst werden kann.

Die Idee ist es nun, diese Zahlenwerte den einzelnen Datenobjekten zuzuordnen, bzw. diese nach und nach aufzusummieren. Das kann für jeden Benutzer einzeln geschehen, oder es ist auch möglich, basierend auf den Interaktionsdaten aller Anwender des Systems, eine globale Wertung der Datenobjekte vorzunehmen.

Der Mehrwert für einen Benutzer von MedioVis äussert sich darin, dass ihm einerseits diese Daten in einer statistisch aufbereiteten Form präsentiert werden können. Das heisst, dass ihm beispielsweise die interessantesten Datenobjekte zu einem bestimmten Themengebiet angezeigt werden können. Andererseits kann er aber auch bei der Analyse einer Treffermenge unterstützt werden, indem in einer Visualisierung Datenobjekte, die schon früher von hohem Interesse waren, speziell markiert werden, um die Aufmerksamkeit des Benutzers auf sich zu lenken.

Neben der Bindung der Interaktionsdaten an einzelne Objekte ist es auch vorstellbar, diese Daten losgelöst von den Datenobjekten an die Ausprägungen einzelner Attribute — also

Attributwerte — zu hängen. Das würde unter anderem das Visualisieren von interessanten Datenbereichen im HyperScatter zulassen, indem unabhängig von der Suchanfrage die Bereiche hervorgehoben werden können, in denen schon zuvor viele interessante Datenobjekte gefunden wurden, also zum Beispiel der Bereich des HyperScatter in dem Filme mit englischem Orginalton verortet sind. Zum Zeitpunkt der Fertigstellung dieser Arbeit kann noch kein konkretes Konzept präsentiert werden, wie Visualisierungen wie der HyperScatter oder die HyperGrid eine solches Interaktions-History umsetzen sollen.

5.4 Anwendungsszenario Teil II (HyperScatter)

Nachdem Kathrin Schanzer im Anwendungsszenario zur klassischen Benutzung von MedioVis (Kapitel 4.1) mit einigen Schwierigkeiten zu kämpfen hatte, bekommt sie nun noch einmal die Möglichkeit, besagte Aufgabe mit dem HyperScatter zu lösen. Aufgabe war es, die neueren Filme von Alfred Hitchcock ausfindig zu machen, welche in der Mediothek in englischem Orginalton zur Verfügung stehen. Dabei trat bei der klassischen Konfiguration von MedioVis mit HyperGrid und ScatterPlot ein Platzproblem auf. Da im ScatterPlot nicht direkt auf die Detailinformationen der einzelnen Datenobjekte zugegriffen werden kann, muss dies über die HyperGrid geschehen. Das heisst, dass der Bildschirmplatz unter beiden Visualisierungen aufgeteilt werden muss. Im Beispiel führte dies schliesslich zu einem für die Benutzerin unlösbaren Problem, weshalb sie sich dazu entschied den arbeitsaufwändigeren und kognitiv belastenderen Weg, unter alleiniger Verwendung der HyperGrid zu gehen.

Nun sitzt Kathrin wieder vor dem Computer und hat die MedioVis-Anwendung gestartet. Sie sucht nach *Hitchcock* und die Suchtreffer werden dieses Mal mit dem HyperScatter dargestellt (siehe Abbildung 37, oben links). Da sie zur Lösung der Aufgabe die Sprache der einzelnen Datenobjekte in Relation mit ihrem Erscheinungsjahr setzen muss, legt sie das Attribut *Language* auf die Y-Achse des HyperScatters (siehe Abbildung 37, oben rechts). Nun sieht sie bereits, welche Datenobjekte in der Mediothek in englischem Orginalton vorhanden sind, denn diese Objekte weisen beim Attribut *Language* einen der beiden Werte *engl.* oder *engl.dt.* auf.

Da sie sich ausschliesslich für diese Datenobjekt interessiert, beschliesst Kathrin den HyperScatter auf diesen Bereich einzuschränken. Dazu zieht sie eine Bounding-Box, welche genau diese beiden Y-Achsen-Abschnitte beinhaltet und drückt auf den erschein-

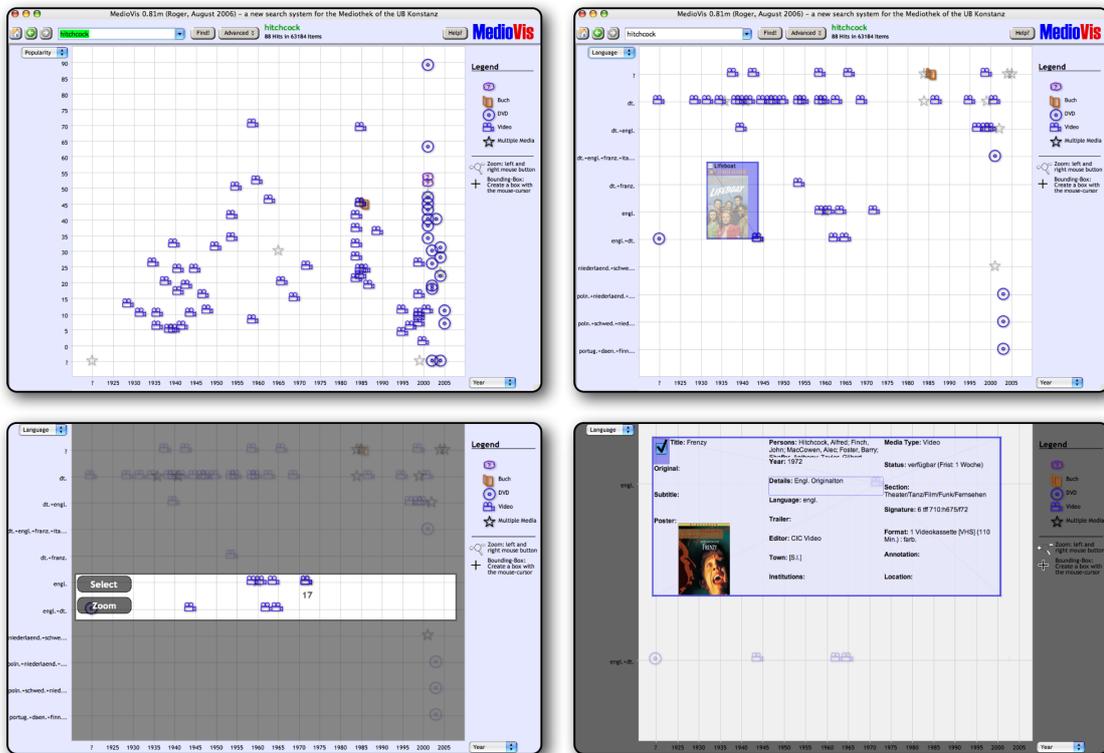


Abbildung 37: MedioVis-Anwendungsszenario mit dem HyperScatter.

den *Zoom*-Button (siehe Abbildung 37, unten links). Es wird in den gewählten Bereich des HyperScatters hineingezoomt und Kathrin kann die neueren Filme, welche sich eher rechts im HyperScatter befinden durch Hineinzoomen in die Detail-Panels der einzelnen Datenobjekt analysieren. So findet sie schnell weitere Filme, die sich gut für ihr Referat eignen würden.

6 Integration des HyperScatters in den MedioVis-Client

In diesem Kapitel soll angesprochen werden, wie die HyperScatter-Visualisierung in den MedioVis-Client integriert werden kann. Dabei wird hier eine eher konservative Methode beschrieben, welche sich im vorhandenen MedioVis-Framework einfach umsetzen lässt. In der Seminararbeit [Dem06] werden weitergehende Konzepte zur Integration des HyperScatters in MedioVis vorgestellt, welche in dieser Arbeit aber aus Platzgründen nicht behandelt werden können.

6.1 Integration durch das Tabbed-Document-Interface-Konzept

Bei der Entwicklung des HyperScatter wurde davon ausgegangen, dass dieser auch im Zusammenspiel mit der in Kapitel 4.2 beschriebenen HyperGrid verwendet wird. Diese beiden Visualisierungen sollen sich gegenseitig so ergänzen, dass einem Benutzer für alle Aufgaben, die sich ihm während der Arbeit mit dem visuellen Suchsystem MedioVis stellen, das optimale Werkzeug zur Verfügung steht.

Wenn für die HyperScatter-Visualisierung ein grosser Bildschirmplatz zur Verfügung steht, kann der Effekt des sich gegenseitigen Überlagerns von Datenpunkten entschärft werden. Je grösser das Platzangebot für den HyperScatter ist, desto weiter liegen auch die darin verorteten Datenpunkte auseinander. Überlagerungen treten vor allem bei einer hohen Dichte von Datenpunkten auf, also wenn für den HyperScatter nur ein kleinerer Platz auf dem Bildschirm zur Verfügung steht, bzw. wenn sehr viele Datenpunkte auf einmal dargestellt werden müssen. Auch bei der HyperGrid wurde in Kapitel 4.1 gezeigt, dass diese zur optimalen Nutzung über viel Platz auf dem Bildschirm verfügen muss. Anders als bei der vorgestellten Kombination von HyperGrid und ScatterPlot soll deshalb hier auf das parallele Bereitstellen beider Visualisierungen als Multiple Coordinated Views verzichtet werden. Statt dessen soll durch Karteireiter zwischen den Visualisierungen umgeschaltet werden (siehe Abbildung 38). Dieses Konzept entspricht dem Tabbed Document Interface, wie es unter anderem von vielen Web-Browsern zur Gruppierung mehrerer Webseiten in einem Anwendungsfenster verwendet wird. Speziell ist dabei, dass über die verschiedenen Tabs nicht auf verschiedene Dokumente zugegriffen wird, sondern dass sich dahinter verschiedene Sichten (Visualisierungen) auf die gleichen Daten befinden.



Abbildung 38: Integration der beiden Visualisierungen HyperScatter (links) und HyperGrid (rechts) in MedioVis mittels dem Tabbed-Document-Interface-Konzept.

Abbildung 38 zeigt zweimal MedioVis: Links mit dem HyperScatter als Visualisierung der Treffermenge, rechts die Visualisierung der gleichen Datenobjekte mit der HyperGrid. Zwischen den beiden Ansichten wird über die Karteireiter (englisch Tabs), welche sich zwischen dem Bereich von Suchformular und Toolbar und dem Visualisierungsbereich befinden, umgeschaltet. Beim Wechseln zwischen HyperScatter und HyperGrid wird so nur die Visualisierung im unteren Bereich des MedioVis-Fensters geändert. Der obere Bereich mit dem Suchformular, der Statusanzeige und der Toolbar wird nicht verändert.

6.2 Koordination der Selektion von Datenobjekten

In Multiple Coordinated Views wird der Status von Datenobjekten zentral gespeichert. Wenn an einer Stelle in der Anwendung ein Datenobjekt fokussiert oder selektiert wird, so werden alle anderen Views aktualisiert und auch dort wird dieses Datenobjekt fokussiert oder selektiert dargestellt. Im vorgeschlagenen Fall werden die beiden Visualisierungen zwar nicht parallel als Multiple Coordinated Views angezeigt, jedoch soll trotzdem bei der Selektion eines Datenobjekts in der einen Visualisierung dieses „global“ selektiert werden. So sind bei jedem Wechsel zwischen der HyperScatter- und der HyperGrid-Ansicht stets die gleichen Datenobjekte selektiert.

Auf diese Weise ist es einem Benutzer zum Beispiel möglich, in der HyperGrid verschiedene Datenobjekte miteinander zu vergleichen, als nächstes zum HyperScatter zu wechseln um

einen ganzen Bereich von Datenobjekten zu selektieren und dann wieder in die HyperGrid zurück zu wechseln um dort die angefangene Analyse fortzusetzen. Da in der HyperGrid keine Selektion von mehreren Datenobjekten auf einmal möglich ist, sondern jedes Objekt einzeln über eine Check-Box selektiert werden muss, kann mit dieser Methode eine solche Aufgabe beschleunigt werden.

Aus Abbildung 38 ist auch ersichtlich, dass sowohl bei der Verwendung des HyperScatters als auch bei Verwendung der HyperGrid die selektierten Datenobjekte in den weiter oben beschriebenen Basket gelegt werden können. Dies geschieht über den Button *Into basket*, welcher unabhängig von der Visualisierung immer angezeigt wird. Dem Benutzer ist es also möglich, sowohl in der Trefferansicht als auch in der Ansicht des Basketes zwischen den beiden Visualisierungen HyperScatter und HyperGrid umzuschalten. Das Umschalten zwischen der Trefferansicht und dem Basket geschieht im Übrigen durch Klicken auf den entsprechenden Button *Results* oder *Basket*, welche sich jeweils rechts oben über den Visualisierungen befindet.

6.3 Konsistenz der Interaktions- und Visualisierungskonzepte

Um den Benutzer von MedioVis dabei zu unterstützen, zwischen den Visualisierungen HyperScatter und HyperGrid ohne grosses Umdenken hin und her wechseln zu können, wird MedioVis-weit ein konsistentes Interaktions- und Visualisierungskonzept angestrebt.

Dies drückt sich zum Beispiel in der gleichen Maustastenbelegung für das Zooming in die Detailansichten (links hineinzoomen, rechts herauszoomen) aus. Oder darin, dass die Attribute der Datenobjekte bei beiden Visualisierungen gleich in Aspect of Interest und Degree of Interest aufgeteilt und dementsprechend in den Detailansichten dargestellt werden.

Es gibt aber auch einige Bereiche, in denen sich die beiden Visualisierungen HyperScatter und HyperGrid gegenseitig noch stärker annähern sollten. So ist zum Beispiel das Medientyp-Symbol im HyperScatter ein wichtiges Element bei der Repräsentation einzelner Datenobjekte, da mit ihm neben den beiden Achsen eine dritte Dimension in den HyperScatter gebracht wird. In der HyperGrid hingegen wird das Medientyp-Symbol noch gar nicht angezeigt. Um das Verständnis beim Benutzer weiter zu verbessern, sollte in jede HyperGrid-Zeile ebenfalls das dem Datenobjekt entsprechende Medientyp-Symbol integriert werden.

7 Zusammenfassung und Ausblick

In dieser Arbeit wurde eine Visualisierung namens HyperScatter vorgestellt, welche zur Darstellung von mehrdimensionalen Daten im Rahmen eines visuellen Suchsystems konzipiert wurde. Der HyperScatter stellt dabei ein leistungsfähiges Visualisierungs- und Analyse-Werkzeug dar, welches vor allem auch unerfahrenen Benutzern die Möglichkeit bieten soll, umfassende Analysen und umfangreiche Recherchen in komplexen Datenräumen durchführen zu können.

Der HyperScatter kombiniert zwei wesentliche Konzepte: Das Zoomable User Interface und die Scatterplot-Darstellung. Durch die Darstellung von Datenobjekten auf einem Scatterplot, liegen die Vorteile des HyperScatters in einer guten Übersichtsdarstellung der visualisierten Treffermengen. Durch das Umsetzen des Zoomable-User-Interface-Ansatzes geht diese Übersicht auch dann nicht verloren, wenn durch semantisches Zooming auf die Detailinformationen einzelner Datenobjekte zugegriffen wird. Problemstellungen wie das direkte Zugänglichmachen von einander überlagernden Datenpunkten, werden im HyperScatter durch verschiedene Interaktions- und Visualisierungskonzepte gelöst, bei denen je nach Bedürfnis semantische oder geometrische Zooming-Verfahren zum Einsatz kommen.

Zur Illustration der konzeptionellen und technischen Machbarkeit wurde der HyperScatter praktisch umgesetzt (stellenweise nur prototypisch) und in MedioVis, einem visuellen Suchsystem, das als Forschungsprojekt an der Arbeitsgruppe Human-Computer Interaction der Universität Konstanz entwickelt wird, integriert.

In dieser Arbeit wurde an mehreren Stellen auf Ideen und konzeptionelle Gedanken hingewiesen, welche mit dem jetzigen Stand der HyperScatter-Implementation noch nicht umgesetzt sind. In einer nächsten Phase, in der die Entwicklung des HyperScatters weitergeführt wird, soll an diesen Punkten angeknüpft werden und die Konzepte konkretisiert und schliesslich in MedioVis und im HyperScatter umgesetzt werden.

Eine weitere wichtige Aufgabe, ist die Evaluation der umgesetzten Konzepte des HyperScatter mittels Benutzer-Tests. Neben Tests im Usability-Labor bieten sich auch Langzeitevaluationen über den Live-Einsatz von MedioVis in der Bibliothek der Universität Konstanz an. Um dies zu ermöglichen muss aber zuerst an den Stellen, an denen der HyperScatter erst prototypisch umgesetzt wurde, noch einige Arbeit investiert werden.

Die aus den Evaluationen gewonnen Erkenntnisse sollen dann in die nächsten Entwicklungsstufen des HyperScatters einfließen.

Literatur

- [AS94a] Christopher Ahlberg and Ben Shneiderman. Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 313–317, New York, NY, USA, 1994. ACM Press.
- [AS94b] Christopher Ahlberg and Ben Shneiderman. Visual information seeking using the filmfinder. In *CHI '94: Conference companion on Human factors in computing systems*, pages 433–434, New York, NY, USA, 1994. ACM Press.
- [AWS92] Christopher Ahlberg, Christopher Williamson, and Ben Shneiderman. Dynamic queries for information exploration: an implementation and evaluation. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 619–626, New York, NY, USA, 1992. ACM Press.
- [BC87] Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.
- [BGR06] Thorsten Buering, Jens Gerken, and Harald Reiterer. User interaction with scatterplots on small screens - a comparative evaluation of geometric-semantic zoom and fisheye distortion. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):829–836, 2006.
- [BR05] Thorsten Buering and Harald Reiterer. Zuiscat: querying and visualizing information spaces on personal digital assistants. In *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, pages 129–136, New York, NY, USA, 2005. ACM Press.
- [Dem06] Mischa Demarmels. Hyperscatt: Ein alternatives visualisierungs- und interaktionskonzept für mediovis. Term Paper, 2006.
- [ERST05] Maximilian Eibl, Harald Reiterer, Peter F. Stephan, and Frank Thissen. *Knowledge Media Design*. Oldenbourg, Sep 2005.
- [FD05] Michael Friendly and Daniel Denis. The early origins and development of the scatterplot. *Journal of the History of the Behavioral Sciences*, 2005.

- [Fur86] George W. Furnas. Generalized fisheye views. In *CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 16–23, New York, NY, USA, 1986. ACM Press.
- [Ger04] Jens Gerken. Evaluation eines metadatenbrowsers - liste vs. leveltable. Bachelor thesis, University of Konstanz, Mar 2004.
- [Gol84] Adele Goldberg. *SMALLTALK-80: The Interactive Programming Environment*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984.
- [HBP02] Kasper Hornbæk, Benjamin B. Bederson, and Catherine Plaisant. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Trans. Comput.-Hum. Interact.*, 9(4):362–389, 2002.
- [HCW88] Don Hopkins, J. Callahan, and M. Weiser. Pies: Implementation, evaluation and application of circular menus. Technical report, University of Maryland Computer Science Department, 1988.
- [Jet03] Hans-Christian Jetter. Usability-evaluation im rahmen von invisip. Bachelor thesis, University of Konstanz, Jul 2003.
- [JGK⁺05] Hans-Christian Jetter, Jens Gerken, Werner A. König, Christian Grün, and Harald Reiterer. Hypergrid - accessing complex information spaces. In *People and Computers XIX - The Bigger Picture, Proceedings of HCI 2005, Vol. 1*. Springer Verlag, Sep 2005.
- [Kön03] Werner A. König. Konzeption und implementation eines 3d-scatterplots zur visualisierung von metadaten. Bachelor thesis, University of Konstanz, Sep 2003.
- [Kön06] Werner A. König. Referenzmodell und machbarkeitsstudie für ein neues zoomable user interface paradigma. Master's thesis, University of Konstanz, Jun 2006.
- [KR05] Peter Klein and Harald Reiterer. The circlesegmentview - a visualization for query preview and visual filtering. In *Proceedings of SPIE Vol. 5669*, pages 327–338. IS and T / SPIE, Jan 2005.
- [KRML03] Peter Klein, Harald Reiterer, Frank Mueller, and Tobias Limbach. Metadata visualization with vismeb. In *7th International Conference on Information Visualisation 2003*, pages 600–605. IEEEComputer Society, Jul 2003.

- [KS96] Eser Kandogan and Ben Shneiderman. Elastic windows: improved spatial layout and rapid multiple window operations. In *AVI '96: Proceedings of the workshop on Advanced visual interfaces*, pages 29–38, New York, NY, USA, 1996. ACM Press.
- [LRKM03] Tobias Limbach, Harald Reiterer, Peter Klein, and Frank Müller. Visualizing metadata: Leveltable vs. granularitytable in the supertable/scatterplot framework. In *Human-Computer Interaction: Theory and Practice (Part II)*, pages 1106–1110. Lawrence Erlbaum Associates, May 2003.
- [MHML05] Linn Marks, Jeremy A.T. Hussell, Tamara M. McMahan, and Richard E. Luce. Activegraph: A digital library visualization tool. *International Journal on Digital Libraries, Special Issue on Information Visualization Interfaces for Retrieval and Analysis*, 5(1):57–69, Mar 2005.
- [NS00] Chris North and Ben Shneiderman. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In *AVI '00: Proceedings of the working conference on Advanced visual interfaces*, pages 128–135, New York, NY, USA, 2000. ACM Press.
- [PF93] Ken Perlin and David Fox. Pad: an alternative approach to the computer interface. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 57–64, New York, NY, USA, 1993. ACM Press.
- [Ras00] Jef Raskin. *The humane interface: new directions for designing interactive systems*. ACM Press. Addison Wesley Professional, New York, NY, USA, 1st edition, mar 2000.
- [RC94] Ramana Rao and Stuart K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 318–322, New York, NY, USA, 1994. ACM Press.
- [RMMH00] Harald Reiterer, Gabriela Mußler, Thomas Mann, and Siegfried Handschuh. Insyder - an information assistant for business intelligence. In *IGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 112–119, New York, NY, USA, 2000. ACM Press.

- [SFRG00] Ben Shneiderman, David Feldman, Anne Rose, and Xavier Ferré Grau. Visualizing digital library search results with categorical and hierarchical axes. In *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, pages 57–66, New York, NY, USA, 2000. ACM Press.
- [Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *VL '96: Proceedings of the 1996 IEEE Symposium on Visual Languages*, volume 00, pages 336–343, Los Alamitos, CA, USA, 1996. IEEE Computer Society.
- [Tuf83] Edward R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, USA, 1983.
- [Wil03] Barry Wilkins. *MELD: A Pattern Supported Methodology for Visualisation Design*. PhD thesis, University of Birmingham, UK, 2003.

Abbildungsverzeichnis

1	Scatterplot von Edmund Halley 1686	7
2	„The power of visual representations“	8
3	FilmFinder	9
4	Geometrischer Zoom	12
5	Semantischer Zoom	12
6	Geometrisch-semantischer Zoom	13
7	Scatterplot, Cluster	14
8	Scatterplot, Multiple Datapoint	15
9	INSYDER, VisMeB, MedioVis	18
10	MedioVis mit eingebundenen externen Datenquellen	22
11	Lageplan	23
12	MedioVis-Client mit HyperGrid und ScatterPlot	24
13	Klassisches MedioVis-Anwendungsszenario	26
14	Aspect of Interest	30
15	Level of Detail	31
16	HyperGrid-Browser	33
17	Spaltenfilter in der HyperGrid	35
18	Extraktion von Detailinformationen aus der HyperGrid	37

19	Der ScatterPlot	38
20	Pie-Menu im ScatterPlot	40
21	Analyse der Treffermenge im HyperScatter	43
22	MedioVis- und HyperScatter-Komponenten	45
23	Darstellung verschieden grosser Treffermengen im HyperScatter	47
24	Automatisches Anpassen der Wertebereiche der HyperScatter-Achsen	49
25	Verortung von Datenobjekt im HyperScatter	51
26	Die verschiedenen Medientyp-Symbole der Mediotheksdaten im HyperScatter	53
27	Auswahl der Achsenbelegung im HyperScatter	54
28	Zoominteraktion mit der Computermaus	57
29	Detail-Zoom eines einzelnen Datenpunktes	59
30	Detail-Zoom eines Multiple Datapoints	62
31	Multiple-Datapoint-Darstellung im HyperScatter	63
32	Die Animator-Klasse	65
33	Geometrisches Zooming im HyperScatter	68
34	Selektion eines einzelnen Datenobjekt im HyperScatter	73
35	Überladen der Bounding-Box zur Selektion mehrerer Datenobjekt	74
36	Druck-Liste von Selektierten Datenobjekt	75
37	MedioVis-Anwendungsszenario mit dem HyperScatter	79
38	Tabbed Document Interface in MedioVis	81