

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit Nr. 213

Visualisierung von Storyboards anhand von Filmdrehbüchern

Paul Kuznecov

Studiengang: Softwaretechnik

Prüfer/in: Prof. Dr. Daniel Weiskopf

Betreuer/in: M. Sc. Markus John,
Dipl.-Ling. Florian Heimerl,
Dipl.-Inf. Kuno Kurzhals

Beginn am: 16. April 2015

Beendet am: 16. Oktober 2015

CR-Nummer: H.5.1, H.5.2, I.7.2

Kurzfassung

Videos sind ein informationsreiches Medium. Gezielte Informationen oder spezielle Szenen zu finden, kann daher mühsam sein. Deshalb ist eine Visualisierung, die Filme zusammengefasst darstellt, wünschenswert. Textuelle Beschreibungen wie Untertitel und Filmdrehbücher können dabei helfen, Filme auf semantischer Ebene zu analysieren. Im Zuge dieser Arbeit werden textuelle Film- und Handlungsbeschreibungen mit Bild- und Multimediakomponenten verknüpft und aufbereitet. Darauf basierend wird eine interaktive Überblicksvisualisierung mit Storyboards entwickelt. Der Inhalt eines Videos wird durch das Konzept von Taglines abstrahiert und in einer erweiterten Timeline-Visualisierung dargestellt. Durch einen interaktiven "Overview + Detail"-Ansatz wird sowohl der Inhalt eines gesamten Films, als auch von einzelnen Szenen visuell repräsentiert, wodurch eine zielgerichtete Analyse ermöglicht wird. Anhand von Anwendungsfällen wird die Flexibilität und Nützlichkeit des Ansatzes aufgezeigt.

Abstract

As videos contain a large amount of information, finding specific scenes or parts can be difficult. To overcome this challenge, a visualization that gives an overview over a movie is highly desirable. Textual descriptions such as subtitles and movie scripts make it possible to analyse movies on a semantic level. In the course of this paper, textual descriptions are coupled with image content to design an interactive overview visualization containing storyboards. The content of a movie is abstracted by the concept of taglines and is visualized in an extended timeline visualization. By using an interactive "overview + detail" approach, an overview of an entire movie as well as of single scenes is given, facilitating a goal-oriented movie analysis. Several use cases demonstrate the flexibility and utility of the devised approach.

Inhaltsverzeichnis

1 Einleitung	11
2 Grundlagen	13
2.1 Zeitliche Videosegmentierung	13
2.2 Visualisierung von zeitbasierten Daten	15
2.3 Textverarbeitung	18
3 Related Work	25
3.1 Klassifizierung von Videovisualisierungen	25
3.2 movieDNA	27
3.3 Visual Storylines	28
3.4 Movie maps	29
3.5 SoundRiver	30
3.6 Weitere Ansätze	31
4 Konzept	33
4.1 Anforderungen	33
4.2 Tagline	34
4.3 Überblicksvisualisierung	35
4.4 Kopplung Bild- und Textinhalt	40
4.5 Manuelle Annotation	41
4.6 Interaktion	42
4.7 Zeitangaben	46
4.8 Filter	47
4.9 Kombination von Taglines	48
4.10 Aufbau der Oberfläche	49
5 Implementierung	51
5.1 Systemübersicht	51
5.2 Farbgebung	52
5.3 Architektur	55
5.4 Datenmodell	56
5.5 Import	57
5.6 Zeitliche Segmentierung	59
5.7 Synchronisation	60
5.8 Evaluation der Synchronisation	65
5.9 Technologien	67

6 Anwendungsfälle	69
6.1 Analyse einer Beziehung von Filmcharakteren	69
6.2 Exploration von Filmstellen zu einem Suchbegriff	73
6.3 Manuelle Annotation	76
7 Zusammenfassung und Ausblick	79
Literaturverzeichnis	81

Abbildungsverzeichnis

2.1	Struktur eines Videos nach Borgo et al. [BCD ⁺ 12]	13
2.2	Zeitpunkt in a) punktbasiertem Zeitmodell und b) intervallbasiertem Zeitmodell. [AMST11]	16
2.3	Intervall in a) punktbasiertem Zeitmodell und b) intervallbasiertem Zeitmodell. [AMST11]	17
2.4	Beispiel für eine Timeline.	18
3.1	movieDNA [PD01] ermöglicht es mit Hilfe des Hierarchical Brushing eine schnelle Übersicht über ein Video zu bekommen, ohne dabei den Kontext des gesamten Videos zu verlieren.	27
3.2	Visual Storylines [CLH12] synthetisiert die wichtigsten Frames eines Videos zu einem zusammengesetzten Bild.	28
3.3	SoundRiver [JBM10] analysiert die Tonspur eines Videos, und bildet die einzelnen Elemente wie Musik, Sprache und Soundeffekte auf visuelle Repräsentationen ab, um eine Überblicksvisualisierung zu generieren.	30
3.4	Moraveji [Mor04] stellt bestimmte Features eines Videos mit farbigen Balken in einer interaktiven Timeline dar.	31
4.1	Eine Tagline fasst semantisch verwandte Tags zusammen.	34
4.2	Das Konzept für die Überblicksvisualisierung. Die drei Ebenen <i>Szenen</i> , <i>Shots</i> und <i>Frames</i> repräsentieren die Struktur eines Videos.	35
4.3	Die MovieView stellt den Inhalt eines gesamten Films auf Szenenebene dar.	36
4.4	MovieView mit visueller Kodierung der relativen Häufigkeit eines Tags durch a) Balkenhöhe und b) Helligkeit.	37
4.5	Die SceneView stellt eine Detailansicht einer einzelnen Filmszene dar.	38
4.6	Die SceneView zusammen mit einem Storyboard bestehend aus neun Frames.	39
4.7	SceneView mit Storyboard bestehend aus a) 3 b) 9 und c) 27 Frames.	39
4.8	Die Kopplung von Bild- und Textinhalt wird durch Interaktion und Synchronisation zwischen dem Videoplayer und dem Filmdrehbuch umgesetzt.	40
4.9	Der Videoplayer mit Kontrolleiste für die manuelle Annotation.	41
4.10	Beim Fahren der Maus über einen Tag in der SceneView erscheint der Frame zur Startzeit des Tags, sowie entsprechende Untertitel oder Notizen.	42
4.11	Interaktion mit einer Aktion des Filmdrehbuchs. Die entsprechende Aktion wird in einem Tooltip angezeigt, gleichzeitig wird die Aktion im Filmdrehbuch markiert und der Videoplayer spielt die entsprechende Filmstelle ab.	43

4.12	Die Auswahl einer Szene in der MovieView. Die ausgewählte Szene wird farblich umrandet und die Szenenbeschreibung erscheint am oberen Rand. Unterhalb der MovieView wird die SceneView der ausgewählten Szene platziert.	44
4.13	Mehrere Szenen können gleichzeitig ausgewählt werden. Die entsprechenden SceneViews werden vertikal gestapelt und durch eine Ziffer gekennzeichnet.	45
4.14	Oben: SceneView im maximierten Zustand ohne Storyboard. Unten: SceneView im minimierten Zustand mit Storyboard. Bei Bedarf kann so eine kompaktere oder detailliertere Darstellung einer Szene dargestellt werden.	45
4.15	Ein Balken in der MovieView (oben) zeigt die aktuelle Position im Video. Die SceneView (unten) zeigt genaue Zeitstempel für den Szenenbeginn, die aktuelle Position, und das Szenenende. Befindet sich der Mauszeiger über einer SceneView, so wird ein Zeitstempel der entsprechenden Zeit im Video angezeigt.	46
4.16	Ausgewählte Taglines in der Liste (links) werden in der MovieView angezeigt.	47
4.17	Die Kombination von Taglines. a) Kombination von Filmcharakteren. b) Kombination von Filmcharakter und Suchbegriff.	48
4.18	Der schematische Aufbau der Oberfläche. Der Überblicksvisualisierung wird der größte Platz zugeordnet. Darunter werden der Videoplayer und das Filmdrehbuch-Panel platziert. Weitere Elemente werden nehmen diesen sinnvoll platziert.	49
5.1	Ein Überblick über das entwickelte System. Als Datengrundlage dient dabei der Film <i>Der Herr der Ringe: Die Gefährten</i>	51
5.2	Jeder Tagline wird eine Farbe zugeordnet. Elemente, die zu der Tagline gehören, werden in der selben Farbe dargestellt.	52
5.3	Die aktuelle Selektion des Benutzers wird durch die Farbe Orange repräsentiert.	53
5.4	Die Architektur des vorgestellten Systems.	55
5.5	Dieses Klassendiagramm modelliert Untertitel, Filmdrehbücher und ihre Relationen. .	56
5.6	Die Struktur einer SRT-Datei.	57
5.7	Auszug vom Filmdrehbuch von <i>Inception</i>	58
5.8	Schematische Darstellung der Szenensegmentierung.	60
5.9	Die Synchronisation des Filmdrehbuchs und der Untertitel. Basierend auf der Synchronisation werden die Taglines der Filmcharaktere erstellt.	64
6.1	Der Dialog für die Kombination von Taglines. Wir wählen "Cobb" und "Mal" aus und bestätigen unsere Auswahl.	70
6.2	Durch die Kombination der Taglines "Cobb" und "Mal" wurde eine neue Tagline "Cobb + Mal" generiert.	70
6.3	Die MovieView für die Charaktere Cobb und Mal aus <i>Inception</i>	71
6.4	Die SceneView der Szene des ersten gemeinsamen Auftretens von Cobb und Mal. A und B zeigen Frames des Storyboard, in dem beide Charaktere zu sehen sind.	71
6.5	Um die Konversation näher zu untersuchen, fahren wir mit der Maus über einen Tag. Ein Vorschaubild mit dem zugehörigen Dialog erscheint daraufhin.	72
6.6	Während der Videoplayer das Video abspielt, bleiben die Visualisierung und das Filmdrehbuch synchron zu einander.	72
6.7	Eine Schlüsselszene in der Beziehung von Cobb und Mal.	73
6.8	Der Benutzer sucht nach dem Begriff "Ring".	74

6.9	Eine neue Tagline namens "Ring" erscheint in der Liste (A) und eine Textstelle die den Suchbegriff enthält wird hervorgehoben (B).	74
6.10	Die resultierende Tagline für den Suchbegriff "Ring" in der MovieView. Die markierten Bereiche A - D stellen besonders relevante Filmabschnitte in Bezug auf den Ring dar.	75
6.11	Ein Überblick des Prologs. A: Die Szenenbeschreibung. B und C: Relevante Frames bezüglich des Rings.	75
6.12	Ein Auschnitt der SceneView des Prologs.	76
6.13	Die Frames mit zugehörigem Untertitel aus der SceneView, in der der Ring erwähnt wird. Hier wird die Geschichte des Rings aus <i>Herr der Ringe</i> beschrieben.	76
6.14	Die Benutzeroberfläche für die manuelle Annotation. A: Der Name der zuerstellenden Tagline. B: Die Startzeit des aktuellen Tags. C: Optionale Notizen zu dem aktuellen Tag. D: Die Endzeit des aktuellen Tags.	77
6.15	Eine neue Tagline wird angelegt und in der MovieView (A), sowie in der aktuellen SceneView (B) angezeigt.	78
6.16	In der maximierten Ansicht der SceneView werden die verfassten Notizen angezeigt.	78

Tabellenverzeichnis

5.1	Die verschiedenen Ähnlichkeitsmaße und ihre Trefferquoten und Genauigkeiten. . .	66
-----	--	----

Verzeichnis der Listings

5.1	Die Implementierung des JFC-Ähnlichkeitsmaßes	63
5.2	Eine verkürzte und vereinfachte Darstellung des Normalisierungsalgorithmus . . .	65

1 Einleitung

Videos sind ein informationsreiches Medienformat. Ihre dynamische Art erlaubt es durch die Verbindung von Bild und Ton hohe Mengen an Informationen anschaulich darzustellen. Doch während die Menge von Videos in unserer zunehmend digitalisierten Welt wächst, steigt der Bedarf für interaktive Systeme, die einen schnellen Überblick geben und eine leichte Navigation von Videos ermöglichen. Die manuelle Analyse von Videos ist ein zeitaufwendiger Vorgang, da sie eine Länge von mehreren Stunden haben können. Eine komplette Betrachtung für eine Analyse wird dann höchst aufwendig, langwierig und fehlerbehaftet.

Webseiten wie IMDB¹ oder Metacritic² enthalten textuelle Zusammenfassungen von Spielfilmen. Durch solche textuellen Beschreibungen kann ein schneller Überblick über die gesamte Handlung eines Films gegeben werden. Spielfilme haben im Unterschied zu normalen Videos den Vorteil, dass in der Regel auch Filmdrehbücher und Untertitel zu diesen existieren. Filmdrehbücher enthalten semantische Beschreibungen wie zum Beispiel detaillierte Szenenbeschreibungen, Dialoge und Informationen über Charaktere eines Films. Für eine genaue Filmanalyse reicht es jedoch nicht aus, schriftliche Zusammenfassungen zu lesen; auch der Bildinhalt muss betrachtet werden. Es erscheint deshalb lohnenswert, diese Informationen eines Filmdrehbuchs automatisiert zu analysieren und mit Bildinhalten zu kombinieren.

Um diese Herausforderungen zu bewältigen, kann eine interaktive Überblicksvisualisierung eingesetzt werden, die den Inhalt eines Films abstrahiert und auf einen Blick darstellt. Um eine effektive und zielgerichtete Analyse zu ermöglichen, sollte diese Überblicksvisualisierung eine Möglichkeit bieten bestimmte Filmabschnitte im Detail zu betrachten und einen Film szenenweise zu analysieren. Damit die Filmanalyse effizient durchgeführt werden kann, sollte eine sinnvolle und hilfreiche Interaktion und Navigation durch einen Film angeboten werden. Das Ziel dieser Arbeit ist es, eine solche Überblicksvisualisierung, basierend auf einer Verknüpfung von Text- und Bildinhalten, zu konzipieren und prototypisch zu entwickeln.

Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Grundlagen: Hier werden die Grundlagen erklärt, die für ein tieferes Verständnis der Konzepte der entwickelten Visualisierung notwendig sind.

¹www.imdb.com

²www.metacritic.com

1 Einleitung

Kapitel 3 – Related Work: Dieses Kapitel beschreibt verwandte Ansätze zum Themengebiet Video-visualisierungen und stellt Anknüpfungspunkte zu dieser Arbeit her.

Kapitel 4 – Konzept: Hier wird das Konzept der Überblicksvisualisierung vorgestellt.

Kapitel 5 – Implementierung: In diesem Kapitel wird das entwickelte System vorgestellt und die Implementierung des Systems beschrieben.

Kapitel 6 – Anwendungsfälle: Hier werden typische Anwendungsfälle beschrieben, die mit Hilfe des entwickelten Systems durchgeführt werden können.

Kapitel 7 – Zusammenfassung und Ausblick: Zum Schluss werden die Ergebnisse der Arbeit zusammengefasst und mögliche Erweiterungen und Ausblicke diskutiert.

2 Grundlagen

2.1 Zeitliche Videosegmentierung

Für eine kompakte Übersichtsvisualisierung ist eine Unterteilung eines Videos in zeitliche Einheiten unabdingbar. Die Unterteilung ermöglicht eine leichtere Navigation des Videos mit Hilfe von Video-browsern [YYL96]. Außerdem erlaubt sie eine Abstraktion über den Inhalt eines Videos zu erstellen, indem beispielsweise Film Inhalte über die einzelnen Segmente aggregiert werden. Ohne jegliche Segmentierung ist es nur möglich, sich auf bestimmte Zeitpunkte eines Videos zu beziehen. Wird ein Video in Segmente unterteilt, so ist es möglich diese zu annotieren, zu benennen oder zu indexieren [KC01]. Eine Videosegmentierung ist daher nützlich.

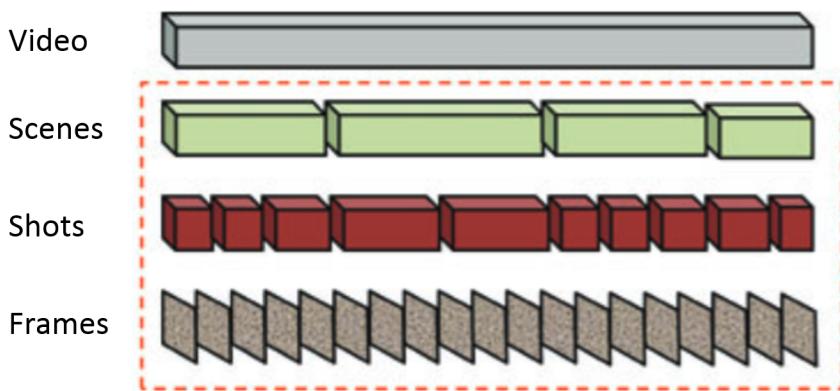


Abbildung 2.1: Struktur eines Videos nach Borgo et al. [BCD⁺12]

Abbildung 2.1 zeigt die Struktur eines Videos. Videos lassen sich in *Frames*, *Shots* und *Szenen* unterteilen [DAE96]. Ein *Frame* ist die elementare Einheit eines Videos und stellt den bildlichen Inhalt zu einem bestimmten Zeitpunkt dar. Ein *Shot* wird als eine Folge von sukzessiven Frames eines Videos definiert, vom Start bis zum Ende eines Filmvorgangs der Kamera. Eine *Szene* ist definiert als Folge von zusammenhängenden Shots. Diese Shots spielen sich am selben Ort ab oder haben einen thematischen Zusammenhang. Szenen haben demnach neben der rein bildbasierten auch eine inhaltliche Bedeutung für das Video.

Die Videosegmentierung stellt Verfahren dar, welche die Einteilung eines Videos in diese Einheiten ermöglichen. Die Aufgabe der automatischen *Shot-Boundary-Detection* ist es für ein gegebenes Video die Shot-Transitionen, also die Zeitpunkte, in denen ein Shot aufhört und der nächste beginnt, aufzufinden.

Im Folgenden werden die wichtigsten Klassen von Shot-Boundary-Detection-Verfahren nach Boreczky und Rowe [BR96] kurz vorgestellt:

Pixeldifferenzverfahren Bei den Pixeldifferenzverfahren werden die Pixel aufeinanderfolgender Frames f und f' betrachtet. Nun wird die Differenz der Frames $d(f, f')$ berechnet, und falls diese einen bestimmten Schwellenwert überschreitet, so liegt eine Shottransition zwischen f und f' vor. Diese Differenz kann beispielsweise berechnet werden, indem die Anzahl der Pixel, deren Wert sich zwischen f und f' mehr als um einen vorgegebenen Schwellenwert verändert. Überschreitet diese Anzahl selbst wiederum einen Schwellenwert, so liegt eine Shottransition vor.

Histogrammverfahren Eine weitere Klasse von Verfahren basiert auf *Histogrammen* [DAE96]. Histogramme sind Häufigkeitsverteilungen von Farbwerten innerhalb eines Frames. Bei diesen Verfahren werden nun Histogramme für aufeinanderfolgende Frames berechnet. Ist die Differenz zwischen den Histogrammen groß genug, so liegt zwischen den Frames eine Shottransition vor. Sei $H(f, k)$ der Wert des Histogramms von Frame f für den Farbwert k . Die einfachste Möglichkeit die Differenz zweier Histogramme zu bestimmen ist nun: $d(f, f') = \sum_{j=0}^N |H(f, j) - H(f', j)|$, wobei N die Anzahl der möglichen Farbwerte eines Pixels ist. Es existieren daneben noch zahlreiche Varianten der Histogrammverfahren, auf die jedoch nicht weiter eingegangen wird. Bei farbigen Videos können sowohl Histogramme der Grauwerte, als auch Histogramme der einzelnen Farbkanäle für die Berechnung verwendet werden.

Bewegungsvektoren Die letzte hier betrachtete Klasse von Verfahren basiert auf dem Prinzip der *Bewegungsvektoren*. Bewegungsvektoren geben an, in welche Richtung und wie sich Pixel zwischen zwei Frames f und f' bewegen. Diese Bewegungsvektoren werden vorwärts und rückwärts, das heißt von f zu f' und von f' zu f berechnet. Innerhalb eines Shots sind die Bewegungen der Frames kontinuierlich, also auch die Richtungen der Pixel ähnlich. Bei einer Shottransition sind die Richtungen jedoch stark verschieden wodurch Shottransitions leicht erkennbar sind [DAE96].

2.1.1 Szenenerkennung

Die Definition einer Szene ist im Vergleich zu der Definition eines Shots eher vage, weshalb eine automatisierte Erkennung von Szenen deutlich schwieriger ist.

Es existieren dennoch bereits Ansätze wie von Rasheed und Shah [RS03], die das Video nicht nur in Shots, sondern auch Szenen unterteilen. Dabei werden erst Shots aus dem Video extrahiert und daraufhin ähnliche Shots zusammengefasst und als Szenen erkannt. Diese Art der Szenenerkennung ist rein bildbasiert und zieht keine semantischen Informationen des Videos in Betracht. In dieser Arbeit wird nicht näher auf diese Verfahren eingegangen, da die Szenenerkennung in dem vorgestellten Ansatz aus Filmdrehbüchern zu den gegebenen Videos extrahiert werden soll.

2.2 Visualisierung von zeitbasierten Daten

Da Filme ein zeitbasiertes Medium sind, ist die Visualisierung von zeitbasierten Daten ein wichtiger Aspekt dieser Arbeit. Um eine nützliche und ansprechende Überblicksvisualisierung zu entwickeln werden die wichtigsten Punkte dieses Visualisierungsbereichs diskutiert.

Bevor mögliche Visualisierungsarten von zeitbasierten Daten beschrieben werden können, muss zuerst der Begriff der Zeit selbst näher beleuchtet werden.

2.2.1 Charakteristika der Zeit

Im Folgenden werden die wichtigsten Eigenschaften der Dimension Zeit nach Aigner et al. [AMST11] erläutert.

Skala

Ordinal Im ordinalen Zeitmodell lassen sich Zeitangaben nur relativ ordnen, also *vor* oder *nach*.

Diskret Im diskreten Zeitmodell existieren diskrete Zeitpunkte einer bestimmten kleinsten Zeiteinheit, wie beispielsweise Sekunden. Hier können Differenzen von Zeitpunkten bestimmt werden.

Kontinuierlich Im kontinuierlichen Zeitmodell existiert zwischen zwei gegebenen Zeitpunkten stets ein weiterer Zeitpunkt. Zeitpunkte können also auf die reellen Zahlen abgebildet werden.

Ausmaß

Punkt-basiert Elemente in punktbasierten Zeitdomänen haben keine zeitliche Dauer und existieren zu einem genauen Zeitpunkt.

Intervall-basiert Elemente in intervallbasierten Zeitdomänen haben eine Start- und Endzeitpunkt und besitzen somit auch eine zeitliche Dauer.

Anordnung

Linear In linearen Zeitmodellen verläuft die Zeit stets vorwärts und es existiert eine Gesamtordnung über die Zeitelemente.

Zyklisch In zyklischen Zeitmodellen existieren stets wiederkehrende Zeitelemente. Dies ist zum Beispiel bei den Jahreszeiten oder den Stunden eines Tages der Fall.

Diese Charakteristika von Zeit beeinflussen die Entscheidung wie zeitabhängige Daten visualisiert werden. In dem Fall von Videos liegt ein diskretes Zeitmodell vor, da ein Video eine kleinste Zeiteinheit besitzt. Ein kontinuierliches Zeitmodell liegt nicht vor, da zwischen zwei gegebenen Frames nicht immer ein weiterer Frame gefunden werden kann. Das Ausmaß bei Videos kann als Intervall-basiert angesehen werden, da die einzelnen Frames eines Videos eine bestimmte Länge, definiert durch die Framerate des Videos, besitzen.

2.2.2 Granularität

Granularität ist eine Art der Abstraktion von Zeit, die dem Menschen den Umgang mit Zeit erleichtern soll [AMST11]. Beispiele für Granularitäten sind Sekunden, Minuten, Tage oder Jahre. Granularitäten beschreiben auch die Beziehung zwischen verschiedenen Zeiteinheiten, wie zum Beispiel die Tatsache, dass eine Stunde aus 60 Minuten besteht.

In einem Zeitmodell können mehrere Granularitäten (Stunden, Tage) auftreten. Bei mehreren Granularitäten müssen Abbildungsvorschriften vorliegen, die diese Granularitäten aufeinander abbilden. Diese können regulär (eine Stunde besteht stets aus 60 Minuten) oder irregulär sein (ein Monat kann 28, 29, 30 oder 31 Tage haben). Es kann aber auch sein, dass nur eine Granularität unterstützt wird. Dann werden Zeitwerte jeweils nur in einer Zeiteinheit wiedergegeben. Zuletzt ist es auch möglich *keine* Granularitäten in einem Zeitmodell zu haben. Dann gibt es keinerlei Abstraktion der Zeit.

Videos haben eine zeitliche Granularität, die durch ihre Framerate definiert wird. Bei einer Framerate von 30 Frames pro Sekunde entspricht die Granularität also $\frac{1}{30}$ Sekunde.

2.2.3 Zeitprimitive

Zeitprimitive sind die grundlegenden Elemente die Daten und Zeit verknüpfen [AMST11]. Die drei wichtigsten Zeitprimitive sind: *Zeitpunkt*, *Intervall* und *Zeitspanne*.

Zeitprimitive können als absolut oder relativ charakterisiert werden. Absolute Zeitprimitive wie der Zeitpunkt und das Intervall beziehen sich immer auf eine konkrete Stelle in der Zeitdomäne. Relative Zeitprimitive wie die Zeitspanne haben keinen absoluten Fixpunkt auf der Zeitachse.



Abbildung 2.2: Zeitpunkt in a) punktbasiertem Zeitmodell und b) intervallbasiertem Zeitmodell. [AMST11]

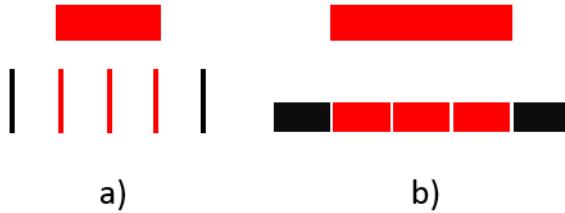


Abbildung 2.3: Intervall in a) punktbasiertem Zeitmodell und b) intervallbasiertem Zeitmodell.
[AMST11]

Ein Zeitpunkt bezieht sich auf einen bestimmten Punkt in der Zeit, abhängig vom Ausmaß des Zeitmodells. In einem punktbasierten Zeitmodell hat ein Zeitpunkt keine Länge, während in einem intervallbasierten Zeitmodell ein Zeitpunkt die Länge der niedrigsten Granularitätsstufe besitzt. Das Zeitintervall ist ein Teil der zugrundeliegenden Zeitdomäne, dessen Start und Ende durch feste Zeitpunkte definiert wird. Durch den Start- und Endzeitpunkt haben Intervalle immer eine genaue Länge. Ein Zeitspanne repräsentiert eine gerichtete Zeitdauer ohne einen fixen Startzeitpunkt. Zeitspannen werden in Einheiten einer bestimmten Granularität ausgedrückt (50 Minuten, 17 Sekunden) und haben im Gegensatz zu Intervallen nicht unbedingt eine klare Länge. So kann die Zeitspanne "2 Jahre" eine Länge von 722 oder 723 Tagen haben, abhängig davon ob eines der Jahre ein Schaltjahr ist.

Zeitprimitive können in Relation zu einander stehen. Welche Relationen dabei möglich sind, hängt von den Arten der Zeitprimitive ab, die verglichen werden. Zwei Zeitpunkte A und B können auf drei verschiedene Arten in Relation zueinander stehen. A ist **vor** B, B ist **vor** A oder A ist **gleich** B. Ein Zeitpunkt A und ein Intervall B können bereits auf acht verschiedene Arten in Beziehung zu einander stehen. A kann zum Beispiel **vor** B, **während** B oder **nach** B sein. A kann außerdem B **starten**, wenn der Startzeitpunkt von B gleich A ist, oder analog kann A B **beenden**. Die restlichen Relationen werden durch die Umdrehung der hier genannten abgedeckt. Zwei Zeitintervalle A und B können bereits auf 13 verschiedene Arten in Relation zu einander stehen, die hier nicht näher beschrieben werden.

Da wir uns in Videos für bestimmte Szenen, Filmstellen oder Ereignisse interessieren, und diese eine zeitliche Länge besitzen, sind im Fall der Videovisualisierung Intervalle das wichtigste Zeitprimitiv.

2.2.4 Timeline

Nachdem nun die wichtigsten Merkmale von Zeit und die verschiedenen Zeitprimitive eingeführt wurden, stellt sich die Frage, wie zeitabhängige Daten visualisiert werden können.

Die Wahl der Visualisierung hängt stark davon ab welche Eigenschaften unsere zeitabhängigen Daten haben und in welchem Zeitmodell wir uns befinden. Auch hängt die Wahl davon ab, welche Art von Aufgaben Benutzer mit einer Visualisierung bewerkstelligen wollen. Wenn die Zeitprimitive einer Visualisierung Intervalle sind, weil zum Beispiel die zugrundeliegenden Datenpunkte feste

2 Grundlagen

Start- und endzeitpunkte haben, so bietet sich eine **Timeline** als Visualisierung an. Die Timeline ist eine simple und intuitive Visualisierung, die es ermöglicht Ereignisse mit einer bestimmten Dauer darzustellen [AMST11]. Dazu werden Intervalle durch Balken oder Linien repräsentiert und entlang einer Zeitachse platziert. Der Beginn einer Linie oder eines Balkens zeigt den Startpunkt an, das Ende entsprechend den Endzeitpunkt. Die zeitliche Dauer eines Intervalls wird durch die Länge repräsentiert.

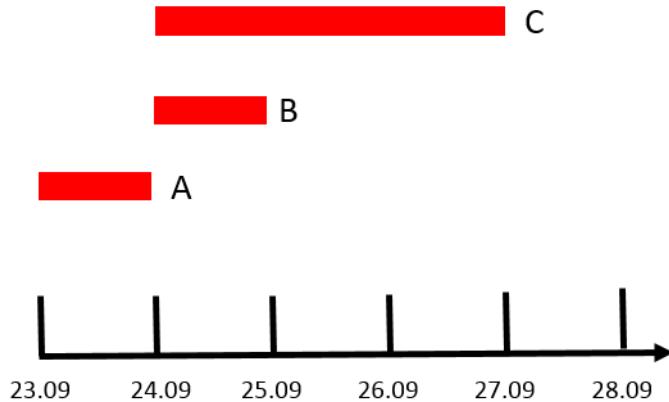


Abbildung 2.4: Beispiel für eine Timeline.

Abbildung 2.4 zeigt ein Beispiel einer Timeline. Hier wird schnell Ersichtlich das Intervall A vor Intervall B stattfindet, da der Rechte Rand von A und der linke Rand von B auf der gleichen horizontalen Position liegen. Auch Längen von Intervallen können anhand der Länge eines Balkens leicht verglichen werden. So ist direkt erkennbar, dass das Intervall C länger ist als Intervall A und, dass Intervall A und B gleich lang sind. Wenn mehrere Intervalle die gleiche Zeitskala haben, können sie in Reihen platziert werden. Das ermöglicht es, die Relationen zwischen Intervallen zu erkennen, wodurch Intervalle von einem Betrachter verglichen werden können. Ein großer Vorteil von Timelines ist ihre Erweiterbarkeit. So können die einzelnen Balken oder Linien weitere Dimensionen der Daten durch zusätzliche visuelle Merkmale darstellen. Dazu kann die Farbe, Höhe oder Umrandung der einzelnen Elemente verändert werden. Auch verschiedene Interaktionsmöglichkeiten können mit einer Timeline umgesetzt werden. Dazu zählt zum Beispiel die Auswahl, das Editieren und Erstellen von Intervallen. [AMST11]

2.3 Textverarbeitung

Um eine Verknüpfung von Untertiteln und Filmdrehbuch herzustellen, müssen verschiedene Techniken der Textverarbeitung eingesetzt werden. Die Herausforderung bei dieser Verknüpfung ist die Überprüfung von Untertiteln und Dialogen auf Ähnlichkeit. Dazu können verschiedene Techniken der Textverarbeitung verwendet werden, die im Folgenden näher betrachtet werden.

Eine mögliche Technik um Texte auf Ähnlichkeit zu überprüfen ist die Kosinus-Ähnlichkeit [AHS08]. Dazu muss ein Vektormodell von Dokumenten erstellt werden, woraufhin die Kosinus-Ähnlichkeit berechnet werden kann. Im Folgenden wird näher auf dieses Vektormodell eingegangen.

2.3.1 Vektormodell

Die Vektorisierung von Dokumenten ist hilfreich, da sie uns einen Zahlenvektor eines Dokuments liefert und es ermöglicht mit Dokumenten zu rechnen und diese mit Techniken der Mathematik zu analysieren. Um aus einem Dokument, oder speziell in dieser Arbeit, einem Untertitel oder Dialog, einen Vektor zu erhalten, werden die sogenannten Termfrequenzen benutzt.

Das hier beschriebene Vektormodell basiert auf Huang [Hua08]. Im Folgenden wird von einer Menge von Dokumenten $D = \{d_1, d_2, \dots, d_n\}$ ausgegangen. Ein Dokument ist dabei definiert als eine Folge von Termen. Als Term wird dabei jedes Wort innerhalb eines Dokuments bezeichnet. Kommt ein Term t in einem Dokument d vor, so gilt $t \in d$. Die Termfrequenz $\text{tf}(t, d)$ eines Terms t innerhalb eines Dokuments d ist die Anzahl der Vorkommnisse von t innerhalb d .

Wir nehmen nun alle distinkten Terme aus D . Für jeden dieser Terme erhalten unsere Dokumentenvektoren einen Zeileneintrag. Die Länge eines Dokumentenvektors ist also gleich der Anzahl aller distinkten Terme. Jeder Eintrag des Dokumentenvektors ist nun gleich der Termfrequenz des entsprechenden Terms innerhalb dieses Dokuments.

Sei also v der Dokumentenvektor von Dokument d und t_i der i -te distinkte Term:

$$(2.1) \quad v : v_i = \text{tf}(t_i, d)$$

Wir haben nun ein Vektormodell für Dokumente definiert und können dieses nutzen, um die Kosinus-Ähnlichkeit von Dokumenten zu berechnen.

2.3.2 Kosinus-Ähnlichkeit

Die Kosinus-Ähnlichkeit [Hua08] ist ein Maß, welches die Ähnlichkeit zweier Vektoren bezüglich ihrer Richtung ausdrückt. Dabei spielt nur der Winkel zwischen den Vektoren eine Rolle, nicht jedoch die Länge der einzelnen Vektoren. Übertragen auf Dokumente wird also berechnet, ob die Sätze sich auf einer oberflächlichen Wortebene ähneln.

Die Kosinus-Ähnlichkeit, oder *Cosine-Similarity*, zweier Vektoren v_1, v_2 wird nun wie folgt berechnet:

$$(2.2) \quad \text{cs}(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|}$$

Dabei ist \cdot das Skalarprodukt und $\| \cdot \|$ die euklidische Norm.

Da die Kosinus-Ähnlichkeit dem Kosinus des Winkels zwischen den beiden Vektoren entspricht, gilt für den berechneten Wert:

$$(2.3) \quad -1 \leq \text{cs}(v_1, v_2) \leq 1$$

Ein Wert von -1 bedeutet, dass die Vektoren in entgegengesetzte Richtungen zeigen, ein Wert von 1 bedeutet, dass sie in dieselbe Richtung zeigen. Ein höherer Wert impliziert also eine höhere Ähnlichkeit beider Vektoren. Da wir nur mit Dokumentenvektoren rechnen, lässt sich der mögliche Wertebereich für die Kosinus-Ähnlichkeit weiter einschränken. Es gilt nämlich, dass die Einträge der Dokumentenvektoren stets nichtnegativ sind, da die Termfrequenz minimal 0 sein kann. Das Skalarprodukt zweier Vektoren mit nichtnegativen Einträgen muss dann folglich ebenfalls nichtnegativ sein. Somit ist auch die Kosinus-Ähnlichkeit nichtnegativ, und wir erhalten stets Werte zwischen 0 und 1.

2.3.3 TF-IDF

Das zuvor vorgestellte Vektormodell, welches lediglich die Termfrequenz für die Einträge des Dokumentenvektors verwendet, hat einen entscheidenden Nachteil. Alle Terme haben die gleiche Gewichtung. Ein Term der in sehr vielen Dokumentenauftritt wird genauso stark gewichtet wie ein Term, der nur in einem Dokument vorkommt. Vor allem bei der Berechnung der Kosinus-Ähnlichkeit ist dies problematisch, da sehr häufige Terme die berechnete Ähnlichkeit im selben Maße beeinflussen, wie seltene Terme. Anhand eines Beispiels lässt sich dieser Sachverhalt gut darstellen:

Gegeben seien die beiden Sätze d_1, d_2 mit den distinkten Termen t_{distinct} :

$d_1 = \text{"The house is nice."}$

$d_2 = \text{"The weather is good."}$

$$t_{\text{distinct}} = [\text{"the"}, \text{"is"}, \text{"house"}, \text{"nice"}, \text{"weather"}, \text{"good"}]$$

$$\text{Es ergeben sich dann die Dokumentenvektoren } v_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \text{ und } v_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

Die Kosinus-Ähnlichkeit ist dann $\text{cs}(v_1, v_2) = 0.5$. Somit haben die beiden Sätze eine relativ hohe Ähnlichkeit, bei einer näheren Betrachtung ist jedoch festzustellen, dass nur die Terme *the* und *is* in beiden Sätzen vorkommen. Diese Terme haben im Allgemeinen eine sehr hohe Vorkommenshäufigkeit und sind unwesentlich für die Semantik eines Satzes. Zwei Dokumente sollten nicht als ähnlich gewertet werden, nur weil solche Terme in beiden Dokumenten vorkommen. Terme wie *house* und *weather* haben dagegen einen größeren Einfluss auf die Semantik eines Satzes und haben im Allgemeinen eine geringere Vorkommenshäufigkeit. Diese Art von Termen sollten somit einen wesentlich höheren Einfluss auf die Ähnlichkeit von Sätzen haben.

Damit seltene Terme stärker gewichtet werden kann das **TF-IDF** (Term Frequency - Inverse Document Frequency) [Ram03] Maß eingesetzt werden. Dieses bezieht sich stets auf einen Dokumentenkorpus. Terme die in vielen Dokumenten vorkommen werden haben einen niedrigen TF-IDF Wert, während Terme die nur in wenigen Dokumenten vorkommen einen hohen Wert erhalten.

Für die Berechnung des TF-IDF wird die IDF (Inverse Document Drequency) eines Terms t im Folgenden definiert. Sei D eine Menge an Dokumenten und sei n die Anzahl der Dokumente aus D , in denen t vorkommt.

$$(2.4) \quad \text{idf}(t, D) = \log\left(\frac{|D|}{n}\right) \quad \text{mit} \quad n = |\{d \in D | t \in d\}|$$

Weil ein Term im Maximalfall in allen Dokumenten auftritt, gilt $n \leq |D|$. Somit gilt für den Bruch $\frac{|D|}{n} \geq 1$. Für einen Term, der in allen Dokumenten auftritt gilt also $\log\left(\frac{|D|}{n}\right) = 0$. In diesem Fall ist die IDF minimal. Dies ist sinnvoll, da ein Term der in allen Dokumenten auftritt nur einen geringen Einfluss auf die Ähnlichkeit zweier Dokumente hat.

Nun lässt sich das TF-IDF Maß für einen Term t und ein Dokument d aus der Dokumentenmenge D wie folgt definieren:

$$(2.5) \quad \text{tf-idf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D)$$

Die Termfrequenz wird also mit der IDF gewichtet.

Wir können nun unser Vektormodell aus Abschnitt 2.3.1 mit Hilfe des TF-IDF verbessern. Dafür definieren wir unseren Dokumentenvektor nun wie folgt:

$$(2.6) \quad v : v_i = \text{tf-idf}(t_i, d, D)$$

Praktisch bedeutet das, dass wir nun nicht nur die Termfrequenz als Vektoreinträge benutzen, sondern diese noch mit der IDF einzelner Terme gewichten.

Berechnen wir nun die TF-IDF Werte der Terme aus dem vorherigen Beispiel.

$$\text{tf-idf}(\text{"the"}, d_1, D) = 1 \cdot \log\left(\frac{2}{2}\right) = 0$$

$$\text{tf-idf}(\text{"the"}, d_2, D) = 1 \cdot \log\left(\frac{2}{2}\right) = 0$$

$$\text{tf-idf}(\text{"is"}, d_1, D) = 1 \cdot \log\left(\frac{2}{2}\right) = 0$$

$$\text{tf-idf}(\text{"is"}, d_2, D) = 1 \cdot \log\left(\frac{2}{2}\right) = 0$$

$$\text{tf-idf}(\text{"house"}, d_1, D) = 1 \cdot \log\left(\frac{2}{1}\right) \approx 0.3$$

$$\text{tf-idf}(\text{"nice"}, d_1, D) = 1 \cdot \log\left(\frac{2}{1}\right) \approx 0.3$$

$$\text{tf-idf}(\text{"weather"}, d_2, D) = 1 \cdot \log\left(\frac{2}{1}\right) \approx 0.3$$

$$\text{tf-idf}(\text{"good"}, d_2, D) = 1 \cdot \log\left(\frac{2}{1}\right) \approx 0.3$$

Die Terme *the* und *is* haben einen TF-IDF Wert von 0, da sie in beiden Sätzen vorkommen. Die anderen Terme haben jeweils denselben Wert von ≈ 0.3 und sind demnach also gleich wichtig.

Nun können wir die Dokumentenvektoren v_1 und v_2 aufstellen und die Kosinus-Ähnlichkeit berechnen:

$$v_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.3 \\ 0.3 \end{pmatrix} \quad v_2 = \begin{pmatrix} 0 \\ 0 \\ 0.3 \\ 0.3 \\ 0 \\ 0 \end{pmatrix} \quad \text{cs}(v_1, v_2) = 0$$

Die Kosinus-Ähnlichkeit hat nach der Einbeziehung des IDF-Werts den Wert 0. Das heißt, dass die beiden Sätze nun als maximal unähnlich behandelt werden. Dies ist eine klare Verbesserung zum berechneten Wert ohne IDF, da die beiden Sätze inhaltlich keine Ähnlichkeiten aufweisen. In einem realen Anwendungsfall enthält der Dokumentenkorpus eine höhere Anzahl an Dokumenten. Hier würden auch die sehr häufig vorkommenden Terme vermutlich nicht in allen Dokumenten vorkommen und somit dennoch einen kleinen Resteinfluss auf die Ähnlichkeit haben.

Ein Nachteil des Vektormodells und der Kosinus-Ähnlichkeit ist die Tatsache, dass die Reihenfolge der Terme innerhalb eines Dokuments nicht berücksichtigt wird. Zwei Dokumente, welche die selben Terme mit den selben Termfrequenzen enthalten haben eine Kosinus-Ähnlichkeit von 1. Speziell bei dieser Arbeit ist dies jedoch nicht optimal, da die Reihenfolge von Wörtern innerhalb eines Satzes für die Synchronisation von Untertiteln und Filmdrehbuch relevant ist.

2.3.4 Weitere Ähnlichkeitsmaße

Neben der Kosinus-Ähnlichkeit mit TF-IDF kann die Ähnlichkeit von Sätzen auch mit anderen Methoden gemessen werden.

Verfahren, welche darauf basieren die Anzahl von Wörtern, die in den zu vergleichenden Sätzen übereinstimmen, zu analysieren, sind sogenannte Word Overlap Measures [AHS08]. Im einfachsten Fall wird die Anzahl an Wörtern, die beide Sätze gemeinsam haben gezählt, und dieser Wert durch die Anzahl aller einzigartigen Wörter beider Sätze geteilt. Der berechnete Wert stellt das Verhältnis von gemeinsamen Wörtern zu der Anzahl aller Wörter dar. Das so erhaltene Ähnlichkeitsmaß ist der Jaccard Similarity Coefficient (JFC) [AHS08].

Sei w_1 die Menge der Wörter von Satz S_1 und w_2 die Menge der Wörter von S_2 . Dann gilt:

$$(2.7) \quad \text{JFC}(S_1, S_2) = \frac{|w_1 \cap w_2|}{|w_1 \cup w_2|}$$

Eine Variation von JFC ist die Folgende:

$$(2.8) \quad \text{JFC}^*(S_1, S_2) = \frac{|w_1 \cap w_2|}{|w_1|}$$

Hier wird die Anzahl der gemeinsamen Wörter mit der Länge von S_1 normalisiert. Im Gegensatz zu JFC führt dies dazu, dass wenn alle Wörter aus S_1 auch in S_2 vorkommen, bereits eine maximale Ähnlichkeit vorliegt. Hier hat die Länge von S_2 keinen Einfluss auf die Ähnlichkeit der beiden Sätze. Auch diese Verfahren haben den Nachteil, dass die Wortreihenfolge nicht berücksichtigt wird.

Zuletzt gibt es noch die linguistischen Ähnlichkeitsmaße [AHS08]. Diese zeichnen sich dadurch aus, dass sie linguistisches Wissen, wie zum Beispiel semantische Beziehungen zwischen Wörtern, für die Berechnung der Ähnlichkeit zweier Sätze einbeziehen. Somit grenzen sich diese Verfahren von den bisher vorgestellten ab, in denen Sätze nur auf rein lexikalischer Ebene betrachtet wurden. Ein Beispiel für ein linguistisches Ähnlichkeitsmaß ist die Sentence Semantic Similarity. Dieses basiert auf dem TF-IDF Maß, jedoch werden die Einträge der Dokumentenvektoren nicht mit dem TF-IDF, sondern mit der maximalen semantischen Ähnlichkeit zu den Wörtern des Ausgangssatzes gewichtet.

2.3.5 Bewertung der Ähnlichkeitsmaße

Nachdem verschiedene Ähnlichkeitsmaße definiert und näher betrachtet wurden, soll nun ihre Eignung für die Synchronisation von Untertiteln und Filmdrehbuch bewertet werden. Dabei spielt auch der Schwierigkeitsgrad der Umsetzung im eigentlichen System eine wichtige Rolle.

Die Kosinus-Ähnlichkeit mit Termfrequenzen ist ein sehr simples Verfahren und kann leicht auf das Problem der Synchronisation übertragen werden. Sowohl Untertitel als auch Dialoge können leicht in Dokumentenvektoren umgewandelt werden, da lediglich die Termfrequenzen der einzelnen Wörter berechnet werden müssen. Ein Nachteil dieses Verfahrens ist jedoch wie bereits erwähnt, dass häufige Terme die berechneten Werte zu stark beeinflussen. Deshalb ist dieses Verfahren für die Synchronisation nur beschränkt geeignet. Wegen der Einfachheit für die Implementierung wird es neben anderen Verfahren aber dennoch im System verwendet.

Die Kosinus-Ähnlichkeit mit TF-IDF als Weiterentwicklung der Variante mit TF bietet eine verbesserte Ähnlichkeitsberechnung, und ist ebenfalls gut auf das gegebene Problem anwendbar. TF-IDF gilt außerdem als ein simples und effektives Verfahren für den Vergleich von Texten [Ram03]. Als Dokumentenkorpus kann hier die Menge der Dialoge im Filmdrehbuch verwendet werden. Vergleicht man nun einen Untertitel mit den Dialogen, berechnet man noch die IDF und bekommt wiederum einen Ähnlichkeitswert. Der Mehraufwand für die Implementierung im Vergleich zu dem Verfahren mit TF ist hier gering. Wegen der gleichzeitig besseren Ergebnisse ist dieses Verfahren ein guter Kandidat für die Synchronisation. Als Nachteil des Verfahrens ist zu werten, dass die Reihenfolge der Wörter innerhalb eines Satzes nicht für die Berechnung des Ähnlichkeitswerts einbezogen wird.

Die Word Overlap Measures sind wegen ihrer einfachen Berechenbarkeit ebenfalls gut für die Synchronisation geeignet. Der Ansatz JFC* normalisiert den berechneten Wert mit der Länge des ersten Satzes. Nimmt man nun den Untertitel als ersten Satz und den Dialog als zweiten, so ist dieses Maß gut geeignet, da Untertitel in vielen Fällen kürzer sind als Dialoge. Kommt der Großteil der Wörter eines Untertitels in einem Dialog vor, so besteht eine hohe Ähnlichkeit, unabhängig von der Länge des Dialogs.

Die linguistischen Ähnlichkeitsmaße haben den Nachteil, dass sie eine höhere Komplexität in der Implementierung haben. Hier müssen Wörter auf einer semantischen Ebene betrachtet werden, sodass die Benutzung von Natural Language Processing (NLP) Tools unabdingbar wird. Das hat den Nachteil, dass diese Tools verstanden und in das System integriert werden müssen.

3 Related Work

Es existiert bereits eine große Vielzahl an verschiedenen Ansätzen für die Zusammenfassung von Videos. Im Folgenden wird zuerst eine Klassifizierung von existierenden Ansätzen der Videovisualisierung vorgestellt und der entwickelte Ansatz in die vorgestellte Taxonomie eingeordnet. Daraufhin werden bestehende Ansätze vorgestellt und deren Relevanz für diese Arbeit diskutiert.

3.1 Klassifizierung von Videovisualisierungen

Borgo et al. [BCD⁺12] stellen eine Taxonomie für Videovisualisierungen auf. Dafür werden vier verschiedene Klassifizierungen definiert. Im Folgenden werden diese beschrieben und der vorgestellte Ansatz in diese Taxonomie eingeordnet.

Es wird zuerst eine grundlegende Einteilung von Visualisierungsansätzen, die Videos als Eingabe nehmen, in die Kategorien *Video-based Graphics* und *Video-Visualization* vorgenommen. Der wesentliche Unterschied dieser Kategorien ist das **Ziel**, welches mit Ansätzen aus den jeweiligen Kategorien erreicht werden soll.

A1. **Video-based Graphics** Der Fokus liegt auf der Manipulation und Darstellung von graphischen Modellen die aus dem Inhalt eines Videos extrahiert wurden. Anwendungsfälle haben oft einen artistischen Hintergrund oder dienen Unterhaltungszwecken.

A2. **Video-Visualization** Das Ziel ist die Unterstützung von Benutzern für eine effiziente und effektive Videoanalyse, indem das mühsame Betrachten eines Videos erleichtert oder gänzlich vermieden wird.

Der entwickelte Ansatz fällt somit klar in die Kategorie Video-Visualization, da der Fokus dieser Arbeit auf einer Überblicksvisualisierung liegt, die eine detaillierte Filmanalyse ermöglicht, ohne einen Film komplett durchschauen zu müssen.

Eine weitere Klassifikation erfolgt nach dem Typen der **Ausgabe** die ein Ansatz verfolgt:

B1. **Weiteres Video oder Animation** Die Ausgabe besteht selbst aus einem Video oder einer Animation. Dieser Fall trifft vor allem bei Video-based Graphics auf.

B2. **Große Bildermenge** Die Ausgabe besteht aus einer großen Bildermenge, die nicht angemessen auf einem Bildschirm angezeigt werden kann. Die einzelnen Bilder können als lineare Folge oder durch Verlinkungen strukturiert sein.

3 Related Work

B3. **Einzelnes zusammengesetztes Bild** Ein einzelnes Bild, bei dem es sich um ein annotiertes Keyframe, oder um eine Zusammensetzung aus Frames von verschiedenen Filmstücken handeln kann, stellt hier die Ausgabe dar.

B4. **Zusätzliche Informationen und Aktion** Die drei vorangehenden Datentypen werden durch weitere Informationen oder Aktionen ergänzt und erweitert. Dabei handelt es sich oft um textuelle oder symbolische Annotationen die Objekte der Ausgabe benennen, beschreiben oder verlinken.

Das entwickelte Konzept fällt klar in die Kategorien B2 und B4. Die Storyboards der Überblicksvisualisierung stellen eine große Bildermenge dar. Durch die Kopplung von Bild- und Textinhalten werden zusätzliche Informationen des Videos ebenfalls dargestellt.

Die dritte Klassifikation betrifft die Art der **Eingabedaten** die ein Ansatz annimmt:

C1. **Originalvideo**

C2. **Zugrundeliegende Modelle**

C3. **Semantische Beschreibungen**

C4. **Bearbeitete Informationen**

C5. **Benutzerinstruktionen** Dazu zählen Befehle und Interaktionen des Benutzers, die Darstellung der Ausgabe beeinflussen.

Auf den entwickelten Ansatz treffen die Kategorien C1, C3 und C5 zu. Ein Hauptaspekt dieser Arbeit ist die Kopplung von textuellen Beschreibungen, wie Filmdrehbüchern und Untertiteln eines Videos, mit dem Bildinhalt. Vor allem Filmdrehbücher stellen dabei eine semantische Beschreibung dar. Durch Interaktion kann der Benutzer außerdem die Ansicht und Darstellung der Überblicksvisualisierung direkt beeinflussen.

Die letzte Klassifikation beschreibt die Ebene der **Automatisierung** eines Ansatzes:

D1. **Überwiegend automatisch**

D2. **Halbautomatisch**

D3. **Überwiegend manuell**

In dieser Arbeit wird ein überwiegend automatischer Ansatz verfolgt, sodass der Benutzer nur die notwendigen Eingabedaten zur Verfügung stellen muss, um eine Visualisierung zu generieren. Die Kopplung von Text- und Bildinhalt wird ohne Einfluss des Benutzers vollautomatisch durchgeführt. Es ist jedoch möglich manuell Annotationen hinzuzufügen oder zu entfernen.

3.2 movieDNA

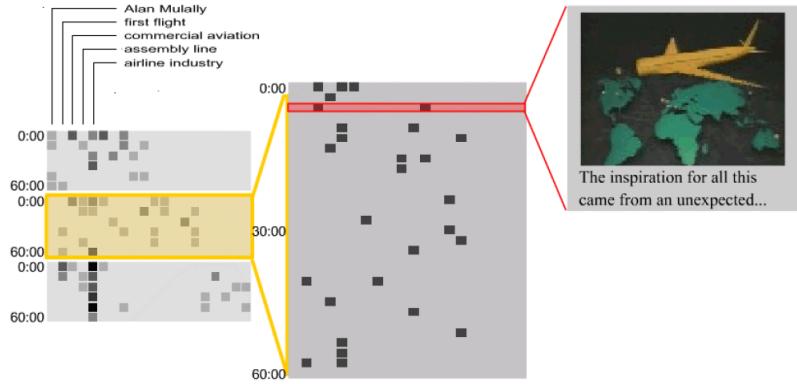


Abbildung 3.1: movieDNA [PD01] ermöglicht es mit Hilfe des Hierarchical Brushing eine schnelle Übersicht über ein Video zu bekommen, ohne dabei den Kontext des gesamten Videos zu verlieren.

Einen interessanten Ansatz verfolgt movieDNA [PD01], dass die Analyse eines Videos durch eine interaktive, hierarchische Überblicksvisualisierung ermöglicht. Bei diesem Ansatz werden Videos zuerst bildbasiert in Segmente unterteilt. Daraufhin wird für jedes Segment die Präsenz eines *Features* berechnet und angezeigt. Features sind dabei Aspekte eines Videos, die für den Benutzer von Relevanz sind, wie zum Beispiel das Auftauchen eines bestimmten Objekts oder das Nennen eines Wortes. Diese Features werden dabei entweder direkt aus dem Video extrahiert oder aus Metadaten zu dem gegebenen Video, wie manuell oder automatisch generierten Transkripten, berechnet. Die Visualisierung ist dabei an ein Gitternetz angelehnt. Jede Reihe repräsentiert dabei ein Segment des Videos und jede Spalte ein Feature. Die Farbe eines Rechtecks in einer bestimmten Spalte und Reihe zeigt an wie stark das entsprechende Feature im jeweiligen Segment vorkommt. Um auch mit langen oder mehreren Videos umzugehen werden Features innerhalb der Segmente aggregiert und ein hierachisches Brushing eingesetzt.

Fährt der Benutzer über die aggregierte DNA, so klappt sich für den gewählten Ausschnitt ein Fenster auf, dass die exakten Vorkommen der einzelnen Features im Detail anzeigt (siehe Abbildung 3.1). In diesem Fenster wird nur noch die An- oder Abwesenheit eines Features angezeigt, und es kann zur genauen Stelle des Videos eines Segments gesprungen werden. Das erlaubt eine fokussierte Analyse eines kleinen Videoabschnitts, während gleichzeitig der Kontext des gesamten Videos beibehalten wird. Ein Einsatzgebiet von movieDNA ist beispielsweise die Analyse von Aufnahmen einer Überwachungskamera, da so schnell interessante Bereiche des Videos aufgefunden werden können, und nicht das gesamte Video durchschaut werden muss. Außerdem kann movieDNA bei Präsentationen mit Slideshows helfen. So kann das Video einer Präsentation mit den einzelnen Slides verknüpft werden.

Der Ansatz des Hierarchical Brushing stellt eine vielversprechende Möglichkeit dar, einen schnellen Überblick eines Videos zu geben, und gleichzeitig eine Detailansicht zu ermöglichen. Das Konzept von

movieDNA hat jedoch auch wesentliche Unterschiede zu dem in dieser Ausarbeitung vorgestellten Konzept. Die Segmentierung des Videos findet nicht auf semantischer Ebene statt, sondern rein auf Bildebene, oder durch eine uniforme Einteilung des Videos in gleichlange Intervalle. Will man nun eine bestimmte Szene eines Filmes analysieren, so müssen alle Segmente durchgeschaut werden. Dies macht eine Analyse auf inhaltlicher Ebene schwierig. Auch ist es nicht möglich mehr als ein Segment gleichzeitig zu untersuchen oder Segmente zu vergleichen. Die in movieDNA angezeigten Features werden vom System automatisch extrahiert. Der Benutzer hat keine Möglichkeit bestimme Features einzublenden, die Reihenfolge der Features zu ändern, oder sogar eigene hinzuzufügen. Vor allem diese Aspekte werden in dem vorgestelltem System verbessert und umgesetzt.

3.3 Visual Storylines



Abbildung 3.2: Visual Storylines [CLH12] synthetisiert die wichtigsten Frames eines Videos zu einem zusammengesetzten Bild.

Einen im Grundsatz verschiedenen Ansatz verfolgt Visual Storylines [CLH12]. Hier wird eine einzelnes, statisches Bild generiert, das versucht den Inhalt eines Videos möglichst gut zu repräsentieren. Dieses Bild wird aus einzelnen Frames des Videos zusammengefügt, die mit Pfeilen verknüpft werden. Die Analyse des Videos erfolgt auf mehreren Ebenen. Zuerst wird das Video in Shots zerlegt, woraufhin diese in Cluster unterteilt werden. Danach werden Relationen zwischen Clustern bestimmt. Dabei benutzt Visual Storylines sowohl Bild- als auch Audioinformationen des Videos. Basierend auf diesen Clustern werden dann die wichtigsten Frames extrahiert und das zusammengesetzte Bild synthetisiert (siehe Abbildung 3.2). Dieser Ansatz funktioniert komplett bild- und audiobasiert, das heißt es werden keinerlei Metainformationen des Videos genutzt.

Ein Vorteil der Visual Storylines ist die Kompaktheit der Visualisierung. Ein einziges Bild stellt den Überblick über das ganze Video da. Der Ansatz ist außerdem sehr generisch und auf jede Art von Video anwendbar. Jedoch bringt der vorgestellte Ansatz auch Probleme mit sich. Aufgrund des bildbasierten

Ansätze können wichtige Inhalte des Videos vom Algorithmus übersehen werden. Die Visualisierung bietet außerdem keine Möglichkeit der Interaktion. Es ist nicht möglich durch das Video zu navigieren oder zu bestimmten Stellen des Videos zu springen.

3.4 Movie maps

Ein Ansatz der das Video sowohl auf Bildebene analysiert, als auch Metadaten und Beschreibungen des Videos einbezieht, ist Movie maps [MT99].

Die Basis von Movie maps bilden die vier folgenden Abstraktionsebenen von Videobeschreibungen:

1. Physical Level
2. Image Level
3. Object Level
4. Discourse Level

Diese Abstraktionsebenen sind hierarchisch aufgebaut, Beschreibungen einer höheren Ebene basieren auf Beschreibungen einer niedrigeren. Beschreibungen des Physical Level betreffen das Speichermedium des Videos selbst, also zum Beispiel das Videoformat oder die Farbkodierung. Diese Beschreibungen sind von geringer Relevanz für die Visualisierung eines Videos. Eine Ebene über dem Physical Level ist das Image Level. Hier wird das Video auf Bildebene betrachtet. Beschreibungen betreffen einzelne Frame des Videos, die beispielsweise durch Bildverarbeitungsalgorithmen generiert werden. Dies können Histogramme, Formen oder Texturen sein. Diese Beschreibungen haben keinerlei Information über die realen Objekte die in den Frames angezeigt werden. Beschreibungen des Object Level haben hingegen Wissen über die Objekte des Videos. Diese können automatisch, wie durch Gesichtserkennungsalgorithmen, generiert werden, oder manuell annotiert werden. Die höchste Abstraktionsebene ist das Discourse Level. Hier betreffen Beschreibungen Metainformationen über den eigentlichen Inhalt des Videos. Die Form und Natur dieser Beschreibungen hängt von der Art und dem Genre des Videos ab. Dazu gehören zum Beispiel Beschreibungen über den Inhalt einer Szene.

Aufbauend auf diesen Abstraktionsebenen wird die eigentliche Visualisierung aufgebaut. Auf dem Image-Level werden OM-Images und Sonogramme als visuelle Darstellung benutzt. OM-Images versuchen die Bewegung und Dynamik eines Objekts des Videos durch eine statische Repräsentation darzustellen. Sonogramme stellen die Tonspur des Videos grafisch dar. Auf dem Object- und Discourse-Level werden Pfeile, Ebenen und eine Variation der Schriftstärke benutzt.

Movie maps ist ein vielschichtiger und komplexer Ansatz. Im Vergleich zu movieDNA und Visual Storylines beschränkt sich Movie maps nicht auf einen rein bildbasierten Ansatz, sondern verbindet diesen mit Metainformationen, Annotationen und Beschreibungen des Videos. Dies erlaubt die Analyse eines Videos auf verschiedenen Ebenen. Außerdem wird dadurch das Problem von bildbasierten Ansätzen vermieden, dass wichtige Teile des Videos übersehen werden. Diese Verknüpfung von semantischen Beschreibungen und Bildinhalten stellt einen interessanten Ansatz dar, der in dieser Arbeit aufgegriffen wird.

3.5 SoundRiver



Abbildung 3.3: SoundRiver [JBMC10] analysiert die Tonspur eines Videos, und bildet die einzelnen Elemente wie Musik, Sprache und Soundeffekte auf visuelle Repräsentationen ab, um eine Überblicksvisualisierung zu generieren.

In den vorhergehenden Abschnitten wurden Visualisierungen betrachtet, die sich auf eine Analyse des visuellen Teil eines Videos konzentrieren. Videos haben aber als audiovisuelles Medium auch die Komponente Ton, die einen wichtigen Teil zum Verständnis eines Videos beiträgt. Der Ton und die Musik, sowie deren Lautstärke beeinflussen die Stimmung eines Videos. Fröhliche Musik deutet häufig auf eine positive Stimmung, während leise, angespannte Musik auf einen Konflikt oder ein aufkommendes Unheil deutet. Somit ist der Ton und die Musik von hoher Bedeutung für eine vollständiges Verständnis eines Videos.

SoundRiver [JBMC10] ist ein Ansatz, bei dem die Tonspur eines Videos analysiert wird, um eine Überblicksvisualisierung zu generieren. Dadurch soll vor allem hörbeeinträchtigten Menschen dabei geholfen werden, die Tonspur eines Videos besser zu verstehen. Am Anfang extrahiert SoundRiver die Tonspur eines Videos und verarbeitet diese. Die Tonspur wird dann in ihrer grundlegenden Einzelteile wie Musik, Soundeffekte und Sprache zerlegt. Das audiovisuellen Mapping bildet diese Einzelteile auf graphische Repräsentationen ab. Dafür werden visuelle Metaphern verwendet. Daraufhin wird in einer visuellen Komposition die eigentliche SoundRiver Visualisierung aus den einzelnen graphischen Bauteilen generiert. In einem letzten, optionalen Schritt kann die Visualisierung durch Verknüpfung mit dem Video und den Untertiteln in einem Videoplayer integriert werden.

Eine Besonderheit von SoundRiver ist, dass es leicht mit anderen Visualisierungen oder Videoplayern verknüpft werden kann. So kann SoundRiver zum Beispiel mit einer Keyframe-Visualisierung kombiniert werden, wobei die einzelnen Frames neben dem SoundRiver platziert werden. Auch die Untertitel werden hierbei mit den Keyframes kombiniert (siehe Abbildung 3.3), um die Visualisierung zu erweitern. In der Visualisierung werden die Frames dann durch Zahlen referenziert. Eine direkte

Verknüpfung mit einem Video ist ebenfalls möglich. Dabei lässt sich das Video normal abspielen, während die SoundRiver Visualisierung synchron zu dem Video mitläuft.

Obwohl SoundRiver ein Ansatz ist, der sich nur auf die Analyse der Tonspur eines Videos beschränkt, ist er im Rahmen dieser Arbeit relevant. So werden in SoundRiver die Elemente der Tonspur auf visuelle Repräsentationen abgebildet, und die daraus resultierende Visualisierung mit dem Video synchronisiert. Im Rahmen dieser Arbeit wird ein ähnlicher Prozess durchgeführt. Hier werden die wichtigsten Elemente eines Filmdrehbuchs (Szene, Dialog, etc.) auf visuelle Repräsentationen abgebildet. Auch wird eine Verknüpfung zwischen Video, Visualisierung und Filmdrehbuch hergestellt. Somit ist SoundRiver ein interessanter Anknüpfungspunkt für diese Arbeit.

3.6 Weitere Ansätze

Moraveji et al. [Mor04] stellen einen Ansatz vor, der auf einer Timeline basiert und bei dem bestimmten Features wie Personen oder Objekten eines Videos eine einzigartige Farbe zugeordnet wird. Diese Farben werden in der Timeline verwendet um für jedes Segment des Videos das wichtigste Feature zu markieren. Benutzer können mit den farbigen Balken interagieren um die entsprechende Kategorie und weitere Informationen. Beim Klick auf einen Balken spielt der integrierte Videoplayer die entsprechende Stelle ab.



Abbildung 3.4: Moraveji [Mor04] stellt bestimmte Features eines Videos mit farbigen Balken in einer interaktiven Timeline dar.

Es wird außerdem eine textuelle Beschreibung des Videos unterhalb des Videoplayers angezeigt. Auch eine Textsuche ist möglich, die Ergebnisse dieser Suche werden mit einer roten Linie in der Timeline markiert. Eine von den Autoren durchgeführte Evaluation durch eine Eye-Tracking Studie belegt, dass die Verwendung von Farben und Linien die Suche nach bestimmten Filmstücken stark vereinfacht.

Die Grundidee dieses Ansatzes Features mit Farben zu kombinieren und als Timeline darzustellen wird für diese Arbeit aufgegriffen und ausgebaut. Eine Limitation ist, dass der Ansatz für besonders lange Videos nicht skaliert, da die Breite der einzelnen farbigen Balken entweder zu gering wird, oder die Segmente des Videos so groß gewählt werden müssen, dass sie nicht mehr sinnvoll eine Filmstelle repräsentieren. Dieses Problem wird in dieser Arbeit mit Hilfe einer hierarchischen Visualisierung gelöst, die es erlaubt sowohl einen Überblick über das gesamte Video, als auch einzelne Szenen im Detail zu bekommen.

4 Konzept

Im Folgenden wird das Konzept der Überblicksvisualisierung vorgestellt, auf dem basierend das System entwickelt wird. Zuerst werden dafür die wichtigsten Anforderungen an die Visualisierung beschrieben.

4.1 Anforderungen

Überblicksvisualisierung Die wichtigste Anforderung an die Visualisierung ist die übersichtliche Darstellung des gesamten Filminhalts. Sowohl Szenen als auch einzelne Shots eines Filmes sollen leicht auffindbar sein. Es soll schnell erkennbar sein, welche Filmcharaktere an einer Szene beteiligt sind und in welchen Szenen ein bestimmter Filmcharakter auftritt. Außerdem soll es möglich sein schnell einen Einblick in den Inhalt von Filmszenen zu erhalten, und bei Bedarf auch eine nähere Analyse durchzuführen. Deshalb muss auch eine Detailansicht einzelner Szenen existieren, die einen genaueren Einblick in eine Szene erlaubt.

Daten Drill-Down Eine weitere wichtige Anforderung an das System ist die Möglichkeit des Daten Drill-Downs. Das bedeutet, dass die Visualisierung stets eine Möglichkeit bieten soll, auf relevante Teile des Videos oder Filmdrehbuchs zu verlinken.

Kopplung Bild- und Textinhalt Eine wichtige Anforderungen an das System ist die Kopplung eines Films, seinen Untertiteln und des dazugehörigen Filmdrehbuchs. Diese soll zum einen auf reiner Datenebene stattfinden, das heißt, dass jeder Untertitel, wenn möglich, zu einem bestimmten Teil des Filmdrehbuchs zugeordnet werden soll. Außerdem soll die Visualisierung diese Kopplung unterstützen und ergänzen. Des Weiteren soll die Möglichkeit bestehen, einen Teil des Filmdrehbuchs auszuwählen und den zugehörigen Filmabschnitt zu betrachten.

Manuelle Annotationen Um eine zielgerichtete Analyse zu unterstützen, soll das System dem Benutzer erlauben, eigene Annotationen hinzuzufügen. Diese sollen dann ebenfalls wie die automatisch generierten Annotationen der Filmcharaktere visualisiert werden. Auch soll es möglich sein Annotationen zu verknüpfen, um beispielsweise alle Szenen zu sehen, in denen mehrere Filmcharaktere interagieren.

Suche Eine Volltextsuche im Filmdrehbuch mit automatischer Verknüpfung und Darstellung in der Übersichtsvisualisierung soll es ermöglichen, schnell bestimmte Filmabschnitte zu einem Suchbegriff aufzufinden.

Interaktion Die Interaktion ist die Grundlage für eine nützliche und hilfreiche Visualisierung [CMS99]. Sie soll eine effektive und intuitive Navigation in der Überblicksvisualisierung ermöglichen und die vorhergehenden Anforderungen sinnvoll unterstützen.

4.2 Tagline

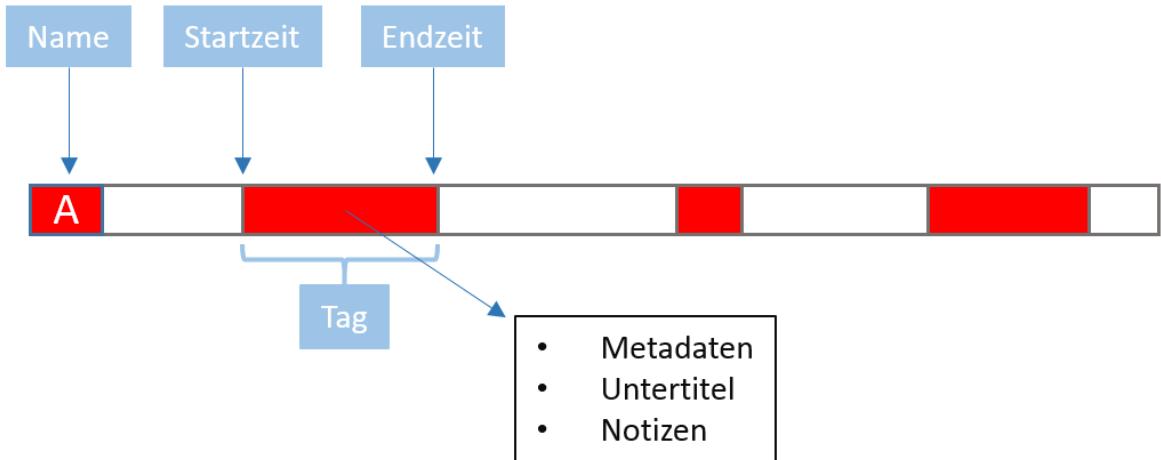


Abbildung 4.1: Eine Tagline fasst semantisch verwandte Tags zusammen.

Um eine Überblicksvisualisierung darzustellen muss der Inhalt eines Films abstrahiert und aggregiert werden. Dafür wird in dieser Arbeit das Konzept der *Tagline* eingeführt.

Eine Tagline (siehe Abbildung 4.1) ist eine geordnete Liste von zusammenhängenden *Tags*. Als Tag wird in dieser Arbeit ein Intervall eines Films bezeichnet, zu dem optional beliebige weitere Metadaten oder Beschreibungen vorliegen. Tags haben eine Start- und Endzeit, und erlauben es sich auf bestimmte Ereignisse, Aktionen oder Dialoge eines Films zu beziehen. Ein Tag kann zum Beispiel ein Untertitel und sein zugehöriger Text sein. Ein Dialog ist ebenfalls ein Tag mit einem Sprecher und dem gesprochenem Text. Auch ein händisch annotierter Filmabschnitt mit Notizen wird als ein Tag angesehen. Eine Tagline fasst nun Tags zusammen die sich auf eine bestimmte Entität beziehen oder eine thematische Ähnlichkeit besitzen. So kann eine Tagline alle Vorkommen eines bestimmten Filmcharakters repräsentieren, oder alle Filmabschnitte in denen eine bestimmtes Objekt angezeigt wird. Jede Tagline hat einen Namen, über den sie referenziert werden kann.

Für die Überblicksvisualisierung auf Szenenebene ist eine Tagline mit den einzelnen Tags zu detailliert und könnte nicht sinnvoll angezeigt werden. Die Aggregation einer Tagline erlaubt es auch lange Videos übersichtlich darzustellen. Deshalb werden die Tags innerhalb einer Szene aggregiert und zu einem Tag zusammengeführt. Für die Überblicksvisualisierung wird nun eine Menge von Taglines ausgewählt, die den Inhalt des Films möglichst gut repräsentiert.

4.3 Überblicksvisualisierung

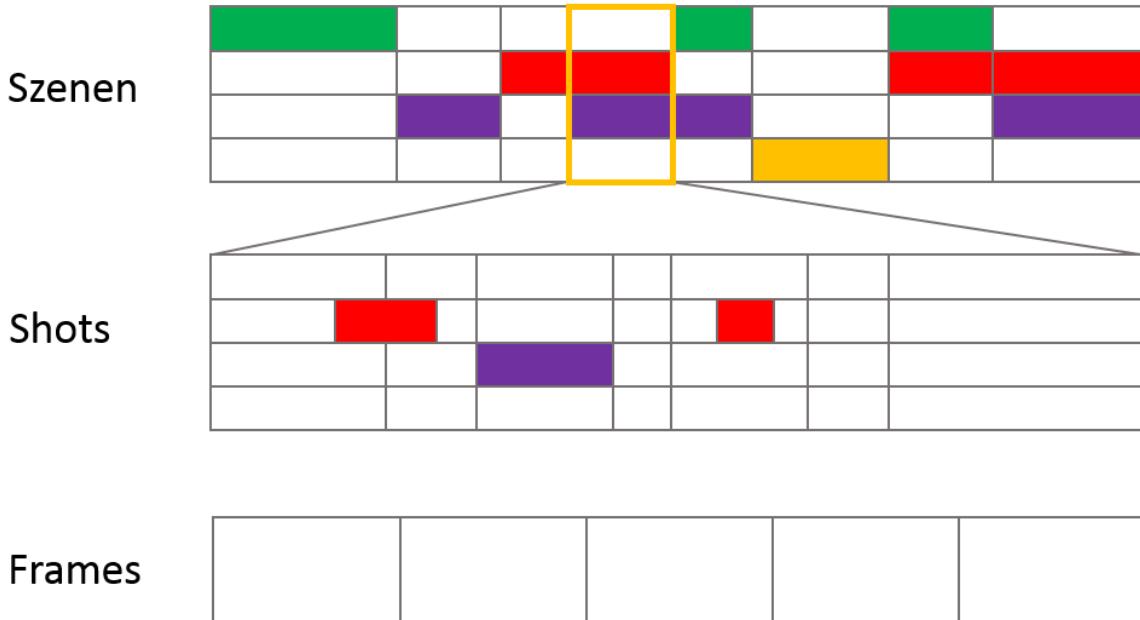


Abbildung 4.2: Das Konzept für die Überblicksvisualisierung. Die drei Ebenen *Szenen*, *Shots* und *Frames* repräsentieren die Struktur eines Videos.

Die Überblicksvisualisierung (Abbildung 4.2) hat einen hierarchischen Aufbau, angelehnt an den Aufbau eines Videos. Videos bestehen wie in Abschnitt 2.1 beschrieben aus Szenen, Shots und Frames. Deshalb ist es sinnvoll, diese verschiedenen Granularitätsstufen auch in der Überblicksvisualisierung visuell darzustellen. Diese werden durch die drei dargestellten Ebenen in Abbildung 5.1 repräsentiert. In der obersten Ebene werden alle Szenen eines Films dargestellt. Dadurch wird ein Überblick über den gesamten Filminhalt gegeben. Für eine detaillierte Analyse kann eine Szene ausgewählt werden, wodurch eine Repräsentation der Szene auf Shotebene angezeigt wird. Gleichzeitig werden auch repräsentative Frames dieser Szene in der untersten Ebene angezeigt, wodurch ein Überblick einer Szene auf Bildebene ermöglicht wird. Die Überblicksvisualisierung folgt somit dem bewährten Prinzip “Overview + Details” [CMS99] der Informationsvisualisierung. Übersicht und Detailansicht werden gleichzeitig angezeigt, wodurch bei einer Detailanalyse der Kontext des gesamten Inhalts erhalten bleibt.

Als Grundlage für die Überblicksvisualisierung wurde eine Timeline ausgewählt. Diese eignet sich aus mehreren Gründen gut für die übersichtliche Darstellung von Filminhalten. Zum einen sind Filme ein lineares und zeitabhängiges Medium. Ereignisse in einem Film haben außerdem einen festen Start- und Endzeitpunkt. Der zeitliche Vergleich von Ereignissen ist ebenfalls ein wichtiger Aspekt in der Filmanalyse. Wie in Abschnitt 2.2 beschrieben sind Timelines eine intuitive und vor allem erweiterbare Möglichkeit, Intervalle darzustellen und sie erlauben einen einfachen zeitlichen

4 Konzept

Vergleich von Intervallen. Auch die Segmentierung eines Films in Szenen und Shots kann mit Hilfe von vertikalen Linien gut in einer Timeline integriert werden.

Im Folgenden werden die wichtigsten Teile und Aspekte der konzipierten Überblicksvisualisierung im Detail beschrieben.

4.3.1 MovieView

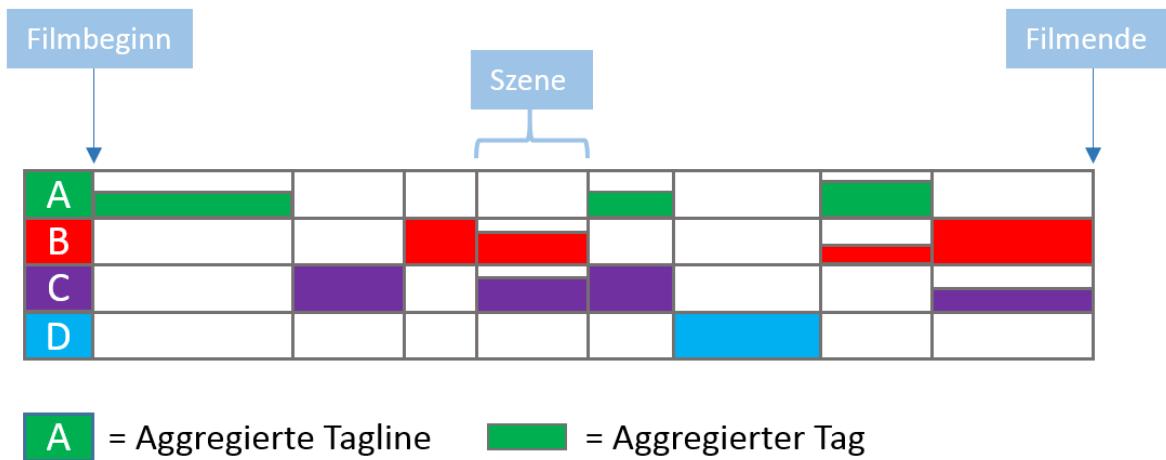


Abbildung 4.3: Die MovieView stellt den Inhalt eines gesamten Films auf Szenenebene dar.

Abbildung 4.3 zeigt die Überblicksvisualisierung auf Szenenebene, im Folgenden nur noch *MovieView* genannt, exemplarisch für einen Film. Die einzelnen Taglines bilden die Basis der Visualisierung. Eine Tagline wird durch eine Reihe von farbigen Balken dargestellt. Die einzelnen Balken stellen die Tags der zugehörigen Tagline dar. Standardmäßig werden hier die fünf wichtigsten Taglines angezeigt. Die vertikalen Linien der MovieView stellen Szenengrenzen dar, welche automatisch durch die in Abschnitt 5.6 beschriebene Analyse der Untertitel und des Filmdrehbuchs extrahiert werden. In der MovieView werden die Taglines auf Szenenebene aggregiert. Dafür wird die Anzahl der Tags in einer Szene betrachtet, die zu einer bestimmten Tagline gehören. Um die relative Häufigkeit einer Tagline innerhalb einer Szene visuell darzustellen, kann zwischen zwei visuellen Kodierungen gewählt werden.

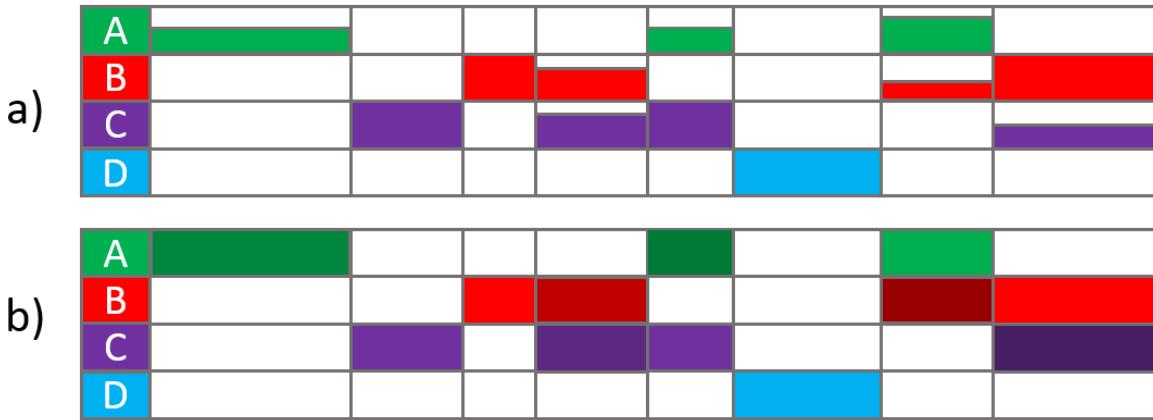


Abbildung 4.4: MovieView mit visueller Kodierung der relativen Häufigkeit eines Tags durch a) Balkenhöhe und b) Helligkeit.

Balkenhöhe Die relative Vorkommenshäufigkeit der Tags einer Tagline innerhalb einer Szene wird auf die Höhe der Balken (siehe a) in Abbildung 4.4) abgebildet. Dadurch entsteht eine Art Balkendiagramm für jede Tagline. Die höchste Vorkommenshäufigkeit wird auf die volle Höhe des Balkens abgebildet. Die kleinste Vorkommenshäufigkeit wird auf eine feste, minimale Höhe abgebildet, sodass der entsprechende Balken erkennbar bleibt. Für alle anderen Szenen wird die Balkenhöhe linear interpoliert.

Der Vorteil dieser Repräsentation ist die Vertrautheit und Verständlichkeit von Balkendiagrammen. Es ist intuitiv, dass Höhe die Häufigkeit widerspiegelt. Außerdem kann die Balkenhöhe von verschiedenen Szenen leicht verglichen werden. Auch Maxima sind durch die volle Balkenhöhe leicht erkennbar. Ein Nachteil der Balkendarstellung ist, dass die einzelnen Balken verschiedene Längen haben. Ein Balken einer langen Szene in der eine Tagline nur durchschnittlich oft vorkommt, hat dadurch einen großen Flächeninhalt. Der Balken einer vergleichsweise kurzen Szene, in der die Tagline aber sehr oft vorkommt, hat einen geringen Flächeninhalt. Dadurch erscheint die erste Szene möglicherweise für einen Benutzer als wichtiger, obwohl eigentlich die zweite Szene eine höhere Relevanz hat.

Helligkeit In diesem Fall sagt die Helligkeit der Farbe eines aggregierten Tags in der MovieView aus, wie wichtig diese Szene im Bezug auf die Tagline ist (siehe b) in Abbildung 4.4). Dem Tag der Szene, welcher die geringste Anzahl an Tags bezüglich einer Tagline hat, wird die dunkelste Farbe zugewiesen. Analog dazu bekommt der aggregierte Tag der Szene mit den meisten Vorkommen die größte Helligkeit. Für alle anderen Werte wird die Helligkeit zwischen dem minimalen und maximalen Helligkeitswert linear interpoliert.

Kommt eine hohe Anzahl von Tags in einer Szene vor, während die meisten anderen Szenen nur wenige Tags enthalten, so wird der aggregierte Tag dieser Szene heller angezeigt. Dadurch können interessante Szenen bezüglich einer Tagline leichter gefunden werden, da sie sich visuell von den anderen unterscheiden. Ein Nachteil dieser Variante ist, dass die Helligkeit von Farben für das menschliche Auge nur schwer vergleichbar ist. Ein heller Balken umgeben von dunklen Balken wird sich zwar leicht absetzen, kann aber nur schwer mit einem hellen

Balken verglichen werden, der weiter entfernt ist. Die Balkenhöhe kann vom menschlichen Auge besser direkt verglichen werden [Mac86].

4.3.2 SceneView

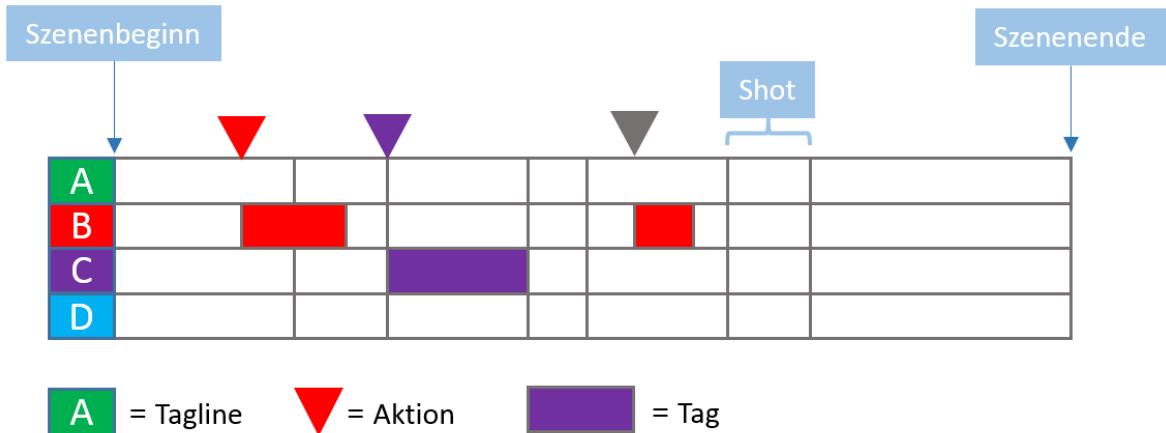


Abbildung 4.5: Die SceneView stellt eine Detailansicht einer einzelnen Filmszene dar.

Wählt der Benutzer eine Szene innerhalb der MovieView aus, so wird eine Detailansicht dieser Szene unterhalb der MovieView angezeigt. Diese Visualisierung einer Szene wird als *SceneView* (Abbildung 4.5) bezeichnet. Diese ähnelt stark der MovieView, repräsentiert jedoch einen kleineren Teil des Films auf einer höheren Granularitätsstufe. Auch hier werden die einzelnen Taglines durch horizontal angeordnete Balken angezeigt. Die vertikalen Linien stellen nun Shotgrenzen und nicht Szenengrenzen dar. Die linke Seite der SceneView entspricht dem Beginn der Szene und die rechte Seite dem Ende.

Neben den Taglines werden auch die im Filmdrehbuch beschriebenen Aktionen in der SceneView dargestellt. Da eine Aktion nicht einfach einem bestimmten Intervall im Film zugeordnet werden kann, werden diese nicht als Tags dargestellt, sondern als farbige Dreiecke am oberen Rand der SceneView. Falls ein Filmcharakter an einer Aktion beteiligt ist, indem er zum Beispiel in dieser erwähnt wird, so wird die Farbe der Tagline des entsprechenden Filmcharakters für das Dreieck gewählt. Als vertikale Position wird hier die Startzeit des Dialogs ausgewählt, der im Filmdrehbuch auf die Aktion folgt. Dies ist eine ungefähre Abschätzung, da sonst keinerlei Informationen zu den Zeitpunkt einer Aktion vorliegen.

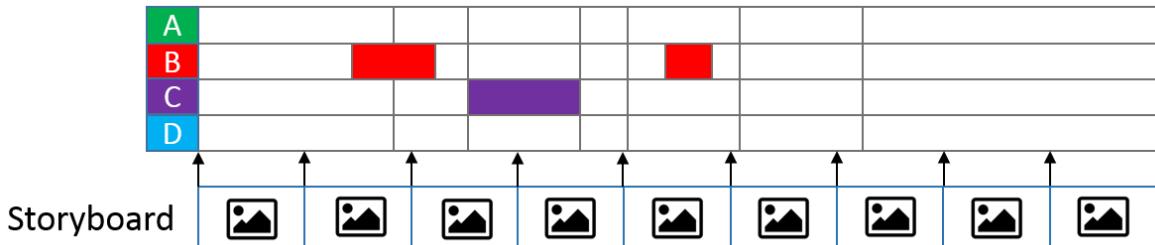


Abbildung 4.6: Die SceneView zusammen mit einem Storyboard bestehend aus neun Frames.

Unterhalb der Taglines wird das Storyboard (Abbildung 4.6) der Szene angezeigt. Dieses dient dazu schnell einen Überblick über den bildlichen Inhalt einer Szene zu bekommen und die Navigation durch eine Szene zu vereinfachen. Hier werden Frames aus der Szene horizontal nebeneinander platziert. Für die Auswahl der Frames wird die Szene in eine beliebige Anzahl gleichlanger Segmente aufgeteilt. Dann wird je ein Frame zu Beginn jedes Segments für das Storyboard ausgewählt. Die Anzahl der Frames ist variabel, hat aber einen Einfluss auf die Kompaktheit der Visualisierung und darauf wie gut eine Szene repräsentiert wird.

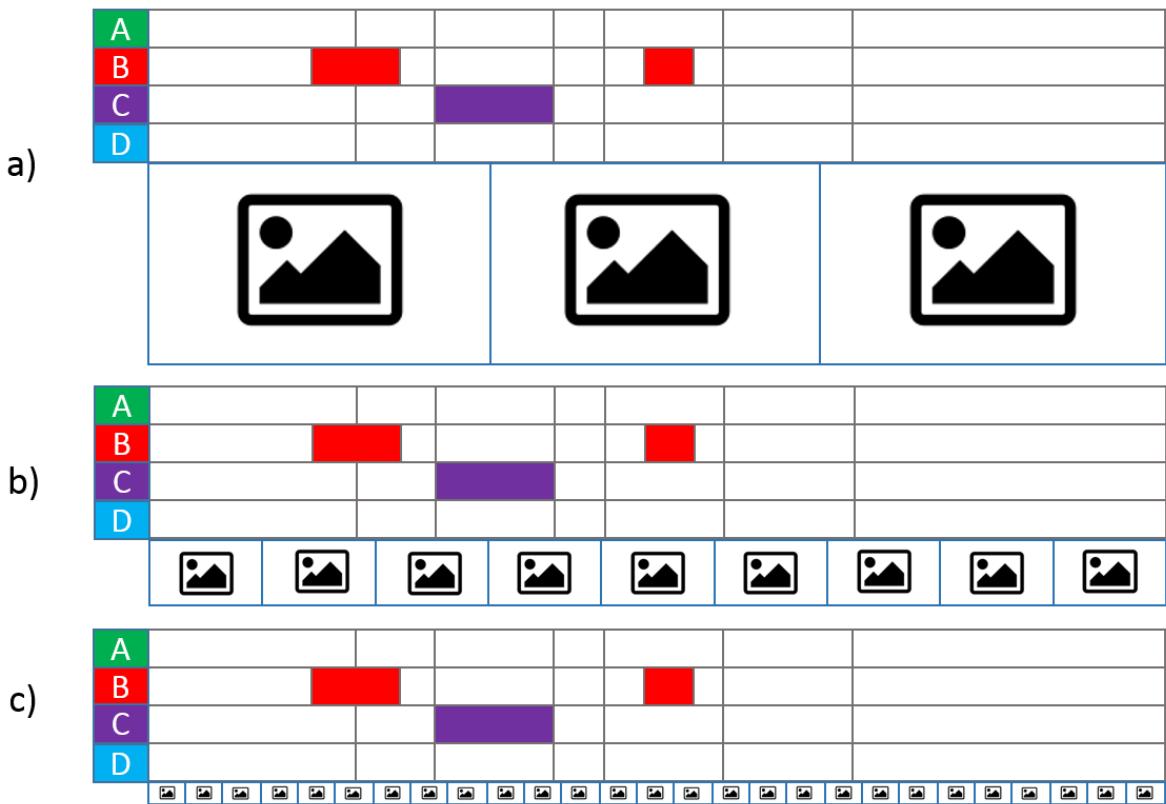


Abbildung 4.7: SceneView mit Storyboard bestehend aus a) 3 b) 9 und c) 27 Frames.

4 Konzept

Wird eine hohe Anzahl an Frames (siehe c) in Abbildung 4.7) gewählt, so kann ein besserer Überblick über den Szeneninhalt gegeben werden, da die Wahrscheinlichkeit steigt, dass ein relevanter Szenenabschnitt durch ein Frame abgedeckt wird. Da die horizontale Ausbreitung der SceneView bei steigender Framezahl aber konstant bleibt, muss jedes Frame verkleinert werden damit alle Frames in einer Reihe platziert werden können. Wird die Anzahl der gezeigten Frames des Storyboards verdoppelt, so ist der Flächeninhalt eines Frames nur noch ein Viertel des vorherigen Flächeninhalts, da sowohl die Höhe als auch die Breite um den Faktor 2 skaliert wird. Dadurch ist der Bildinhalt der einzelnen Frames schwerer zu erkennen. Gleichzeitig führt dies aber auch zu einer Verringerung der vertikalen Ausbreitung der SceneView. Dies wiederum führt zu einer kompakteren Repräsentation einer Szene, was es ermöglicht mehr Szenen gleichzeitig anzuzeigen. Bei einer geringen Anzahl an Frames (siehe a) in Abbildung 4.7) wächst die Höhe der SceneView und es können nur wenige Szenen gleichzeitig angezeigt werden. Im Gegenzug ist es möglich die einzelnen Frames genauer zu betrachten und sie sind leichter erkennbar.

Aus diesen Gründen wurde die Anzahl der Frames nicht auf einen bestimmten Wert fixiert, sondern lässt sich von dem Benutzer individuell einstellen. In der Standardansicht werden neun Frames (siehe b) in Abbildung 4.7) angezeigt, was erfahrungsgemäß einen guten Kompromiss aus Übersichtlichkeit und Erkennbarkeit der Frames darstellt.

4.4 Kopplung Bild- und Textinhalt

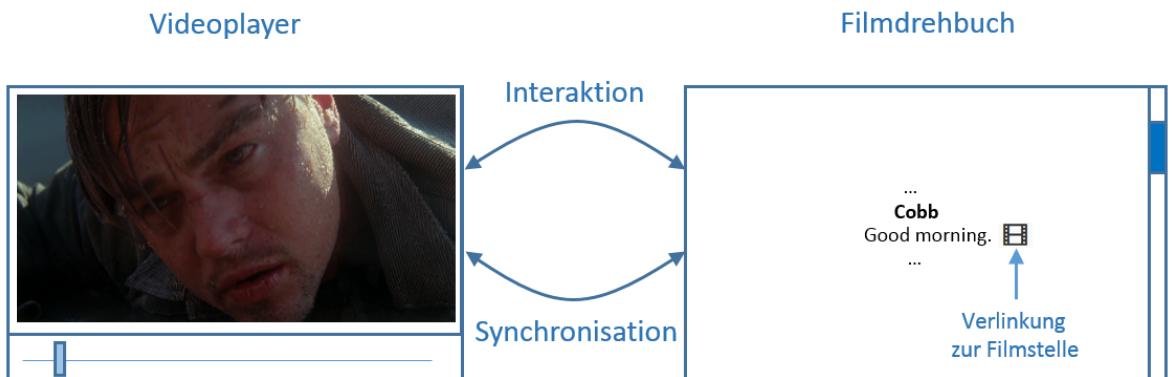


Abbildung 4.8: Die Kopplung von Bild- und Textinhalt wird durch Interaktion und Synchronisation zwischen dem Videoplayer und dem Filmdrehbuch umgesetzt.

Im Folgenden wird das Konzept für die Kopplung von Bild- und Textinhalten beschrieben. Die Kopplung wird durch die beiden Konzepte Interaktion und Synchronisation umgesetzt. Bei der Interaktion wird die Kopplung von dem Benutzer angestoßen. Wurde für einen Teil des Drehbuchs eine entsprechende Filmstelle gefunden, so erscheint ein Filmsymbol (siehe Abbildung 4.8) neben der entsprechenden Stelle des Drehbuchs. Wird dieses Symbol ausgewählt, so spielt der Videoplayer diese Stelle im Video ab. Wenn der Benutzer mit Hilfe des Videoplayers zu einer bestimmten Filmstelle

navigiert, so wird, wenn möglich, die entsprechende Filmstelle im Filmdrehbuch markiert. Somit besteht eine bidirektionale Verknüpfung zwischen dem Videoplayer und dem Filmdrehbuch. Die Synchronisation läuft automatisch ohne Einfluss des Benutzers ab. Während das Video abgespielt wird, wird stets der entsprechende Teil des Drehbuchs farblich hervorgehoben und das Filmdrehbuch-Panel scrollt zu der jeweiligen Stelle im Drehbuch.

4.5 Manuelle Annotation

Für die manuelle Annotation wird das Konzept der Tagline (siehe Abschnitt 4.2) aufgegriffen. Eine Sammlung von zusammengehörenden Annotationen wird dabei als Tagline angesehen. Die Tags dieser Tagline sind die Filmstellen, die vom Benutzer annotiert wurden. Durch Notizen werden Tags Meta-informationen zugeordnet. Somit können verwandte Filmstellen in einer Tagline zusammengefasst werden.

Da Annotationen des Benutzers auch Taglines sind, können diese ebenfalls in der MovieView und SceneView angezeigt werden. Somit werden die eigenen Annotationen sinnvoll in die bestehende Überblicksvisualisierung integriert. Die manuelle Annotation von Filmstellen wird über eine Benutzeroberfläche ermöglicht, die mit dem Videoplayer verbunden ist.



Abbildung 4.9: Der Videoplayer mit Kontrolleiste für die manuelle Annotation.

Der Videoplayer hat dabei einen simplen Aufbau. Unterhalb des eigentlichen Videos befinden sich die Kontrollelemente (siehe Navigation in Abbildung 4.9), mit denen die Navigation durch das Video ermöglicht wird. Ein Slider erlaubt es an beliebige Stellen des Videos zu springen. Rechts daneben wird

4 Konzept

der aktuelle Zeitpunkt des Videos dargestellt. Die „<“ und „>“-Knöpfe erlauben es, sich einen Frame vor beziehungsweise zurück im Video zu bewegen. Das ist vor allem für die manuelle Annotation nützlich, da Filmstellen so mit einer hohen Präzision ausgewählt werden können. Die „«“ und „»“-Knöpfe ermöglichen es, zum nächsten beziehungsweise letzten Shot bezüglich des aktuellen Zeitpunkts zu springen. Das erlaubt es effizient durch eine Szene oder einen größeren Teil des Videos zu navigieren und dennoch sicher zu sein, den Teil des Videos, für den sich der Benutzer interessiert, nicht zu überspringen, da die Frames eines Shots in der Regel einen ähnlichen Bildinhalt haben.

Unter dem Videoplayer befindet sich die Kontrollleiste für die manuelle Annotation (siehe Annotation in Abbildung 4.9). In der obersten Reihe befindet sich ein Texteingabefeld, über den der Name der eigenen Tagline gesetzt wird. Der Knopf „Ende“ wird betätigt, sobald der Benutzer eine Filmstelle mit Start-, -endzeitpunkt und einer optionalen Notiz erzeugt hat. Will der Benutzer einen neuen Tag anlegen, so drückt er den „Start“-Knopf. Dadurch wird die Startzeit des aktuellen Tags gesetzt. Analog wird für die Endzeit verfahren. Eine Dropdown-Liste erlaubt es einen vordefinierten Typen für den Tag zu wählen. Die Pfeiltasten auf der linken und rechten Seite navigieren zum vorherigen beziehungsweise nächsten Tag. Ein Eingabefeld am unteren Rand erlaubt dem Benutzer beliebige Notizen zu einem Tag zu schreiben.

Eine genaue Beschreibung der manuellen Annotation und wie diese mit dem Rest des Systems interagiert, wird mit Hilfe eines Anwendungsfalls in Abschnitt 6.3 erklärt.

4.6 Interaktion

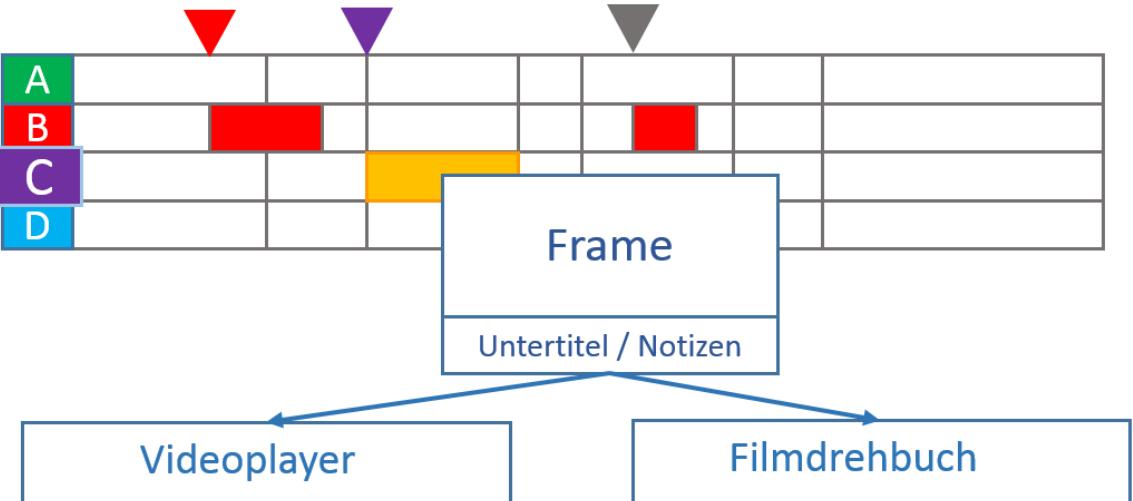


Abbildung 4.10: Beim Fahren der Maus über einen Tag in der SceneView erscheint der Frame zur Startzeit des Tags, sowie entsprechende Untertitel oder Notizen.

Im Folgenden wird das Konzept für die Interaktion der Überblicksvisualisierung beschrieben. Ein besonderer Fokus wurde auf die Interaktion innerhalb der SceneView gelegt. Diese bietet zahlreiche

interaktive Möglichkeiten für die Verknüpfung mit dem Video und Filmdrehbuch. Um eine Synchronisation zwischen Visualisierung, Video und Filmdrehbuch zu realisieren, sind die einzelnen Tags der SceneView interaktiv. Wählt der Benutzer einen dieser Tags mit der Maus, so spielt der Videoplayer die Filmstelle an der Startzeit des gewählten Tags. Wenn der Tag zu einer bestimmten Stelle des Drehbuchs gehört, so wird dieser Teil in dem Filmdrehbuch-Panel hervorgehoben. Dies ist beispielsweise dann der Fall, wenn für einen Untertitel automatisch der korrekte Dialog im Filmdrehbuch gefunden wurde.

Gleichzeitig erscheint beim Fahren der Maus über einen Tag ein Vorschaubild, welches den Frame des Films zur Startzeit des Tags anzeigt (Abbildung 4.10). Der Tag der entsprechenden Tagline wird orange hervorgehoben und der Name der Tagline wird vergrößert dargestellt, damit schnell erkennbar ist welcher Tag ausgewählt wurde und zu welcher Tagline er gehört. Unterhalb des Vorschaubildes wird der Text angezeigt, der mit dem Tag verknüpft ist. Handelt es sich um einen Dialog, so wird auch der Sprecher angegeben. Bei einem Untertitel handelt es sich um den Untertiteltext, bei einem benutzerdefinierten Tag stehen hier die Notizen.



Abbildung 4.11: Interaktion mit einer Aktion des Filmdrehbuchs. Die entsprechende Aktion wird in einem Tooltip angezeigt, gleichzeitig wird die Aktion im Filmdrehbuch markiert und der Videoplayer spielt die entsprechende Filmstelle ab.

Außerdem kann auch mit den Aktionen einer Szene ähnlich wie mit den Tags interagiert werden. Fährt der Benutzer mit dem Mauszeiger über eine Aktion, so wird diese farblich hervorgehoben und es erscheint ein Tooltip, der den Text dieser Aktion anzeigt (siehe Abbildung 4.11). Klickt der Benutzer auf eine Aktion, so springt der Videoplayer wie gewohnt zu dem Startzeitpunkt der Aktion und der entsprechende Teil des Filmdrehbuchs wird im Filmdrehbuch-Panel farblich hervorgehoben.

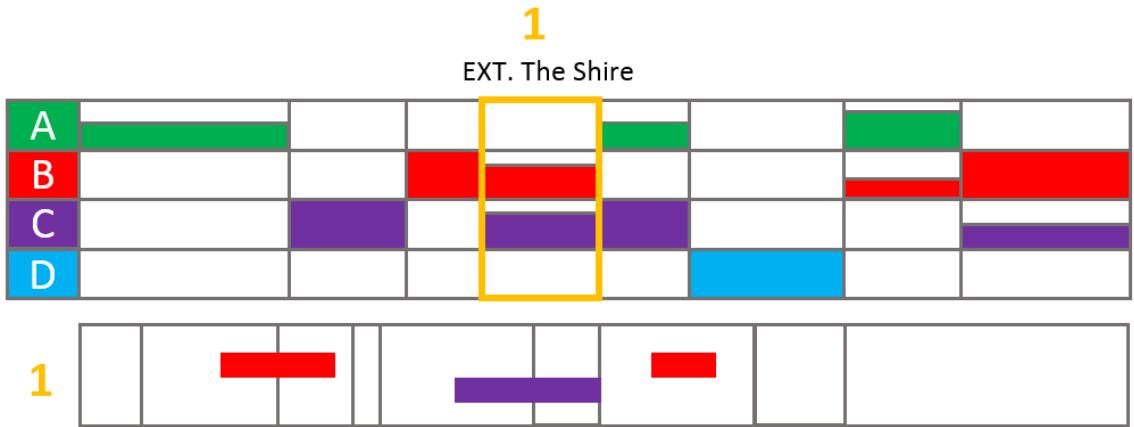


Abbildung 4.12: Die Auswahl einer Szene in der MovieView. Die ausgewählte Szene wird farblich umrandet und die Szenenbeschreibung erscheint am oberen Rand. Unterhalb der MovieView wird die SceneView der ausgewählten Szene platziert.

Die vertikalen Trennlinien der MovieView stellen Szenengrenzen dar. Zwei aufeinanderfolgende Trennlinien schließen somit eine Szene ein. Eine SceneView wird ausgewählt, indem der Benutzer mit der Maus über den rechteckigen Bereich einer Szene in der MovieView fährt. Daraufhin erscheint die Beschreibung der Szene am oberen Rand der MovieView (siehe Abbildung 4.12) um einen ersten, kurzen Einblick in die Szene zugeben. Klickt der Benutzer nun in diesen Bereich, wird das entsprechende Rechteck der Szene in der MovieView farblich umrandet. Es erscheint auch, wie in Abbildung 4.12 zu sehen, eine Zahl über diesem Rechteck. Diese erscheint ebenfalls am linken Rand der nun erschienen SceneView. Die Zahl dient der eindeutigen Referenzierung von SceneViews zu Szenen der SceneView, da sonst bei mehreren angewählten Szenen schnell der Überblick verloren gehen kann, welche SceneView zu welcher Szene gehört. Die Zahlen werden sequentiell, beginnend bei 1 vergeben, in der Reihenfolge, in der die Szenen ausgewählt wurden.

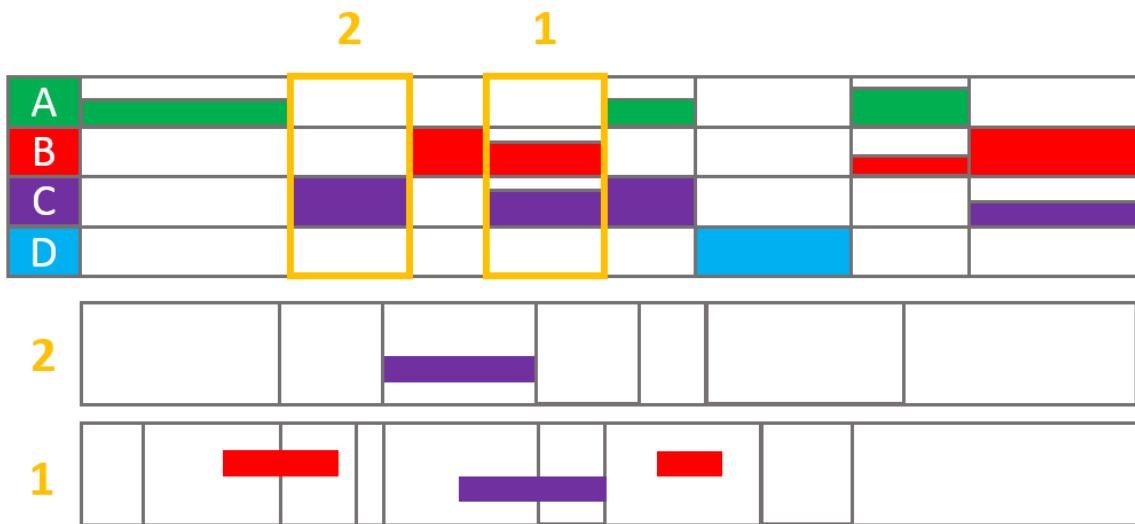


Abbildung 4.13: Mehrere Szenen können gleichzeitig ausgewählt werden. Die entsprechenden SceneViews werden vertikal gestapelt und durch eine Ziffer gekennzeichnet.

Um den Vergleich von Szenen zu ermöglichen erlaubt die MovieView eine Multiselektion (siehe Abbildung 4.13) von SceneViews. Wählt der Benutzer nun einen weiteren Szenenbereich in der MovieView aus, so wird diese Szene zusätzlich selektiert und es erscheint eine weitere SceneView. Diese wird als oberste SceneView platziert, die anderen SceneViews werden dafür entsprechend nach unten verschoben. Dies wird dadurch begründet, dass die als letztes gewählte Szene vermutlich für den Benutzer am relevantesten ist.

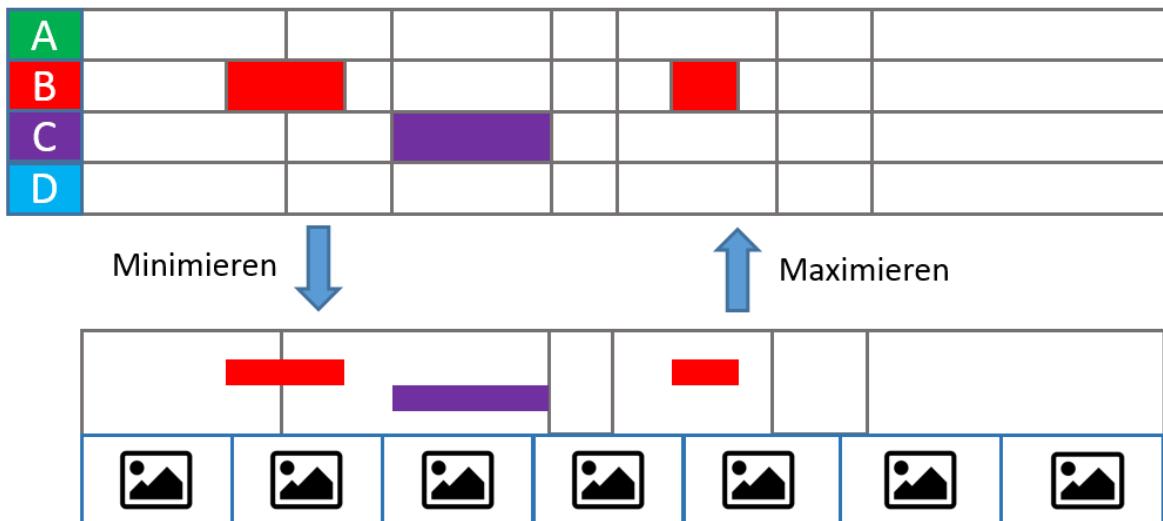


Abbildung 4.14: Oben: SceneView im maximierten Zustand ohne Storyboard. Unten: SceneView im minimierten Zustand mit Storyboard. Bei Bedarf kann so eine kompaktere oder detailliertere Darstellung einer Szene dargestellt werden.

4 Konzept

Der Benutzer kann die SceneView minimieren oder maximieren. Im minimierten Zustand (unten in Abbildung 4.14) werden die einzelnen Balken der Taglines verkleinert dargestellt. Die Namen der Taglines werden in dieser Ansicht nicht angezeigt. Dadurch wird die SceneView in vertikaler Richtung kompakter, was es erlaubt mehr Szenen gleichzeitig durch SceneViews anzuzeigen, da diese vertikal gestapelt werden. Im minimierten Zustand werden auch keine horizontalen Trennlinien zwischen den Taglines gezeichnet, da bei der geringen Höhe der Balken diese schwer zu erkennen wären. Außerdem kann der Benutzer das Storyboard unterhalb der SceneView ein- und ausblenden (siehe Abbildung 4.14). Dadurch kann die Kompaktheit der Visualisierung bei Bedarf noch weiter erhöht werden und es können noch mehr SceneViews gleichzeitig betrachtet werden.

4.7 Zeitangaben

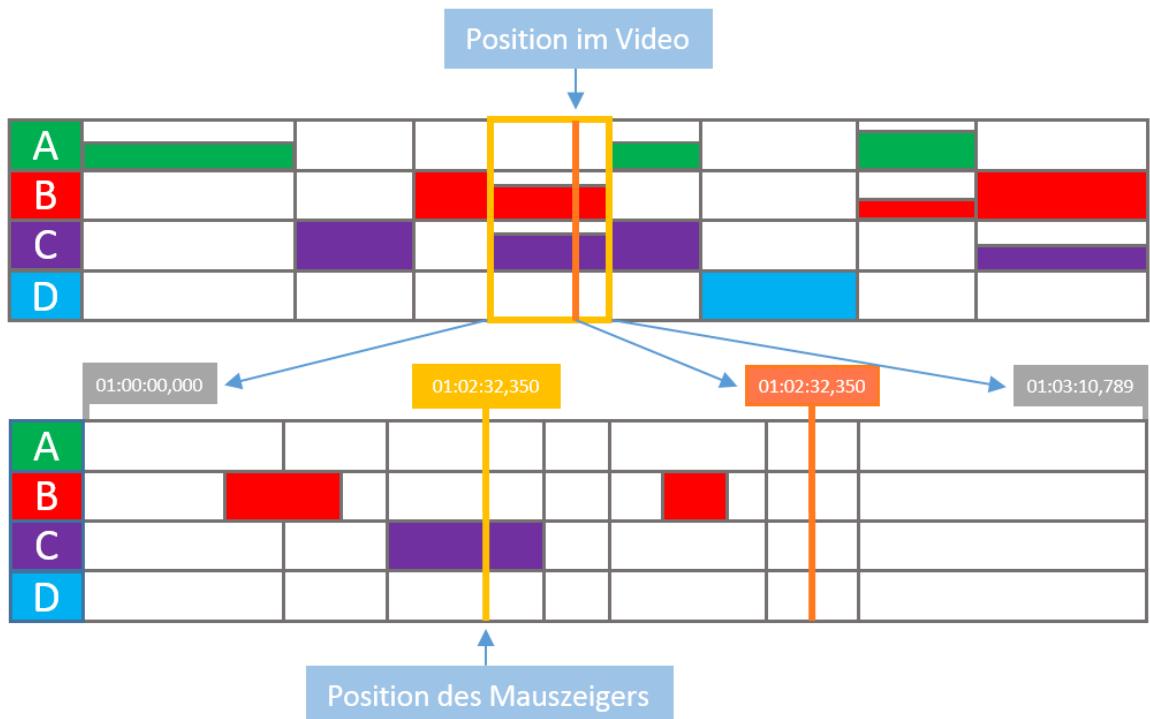


Abbildung 4.15: Ein Balken in der MovieView (oben) zeigt die aktuelle Position im Video. Die SceneView (unten) zeigt genaue Zeitstempel für den Szenenbeginn, die aktuelle Position, und das Szenenende. Befindet sich der Mauszeiger über einer SceneView, so wird ein Zeitstempel der entsprechenden Zeit im Video angezeigt.

Die Zeit ist bei Videos als lineares Medium ein wichtiger Aspekt. Die Visualisierung soll daher dem Nutzer stets einen zeitlichen Kontext geben, damit dieser weiß wo er sich zeitlich im Video befindet und zu welchen Zeitpunkten oder Intervallen bestimmte Ereignisse stattfinden. Diese Zeitangaben werden in die hierarchische Struktur der Überblicksvisualisierung eingebettet (siehe Abbildung 4.15).

Die MovieView zeigt durch einen vertikalen Balken die aktuelle Position des Video. Dadurch wird eine globale Übersicht gegeben, an welchem Zeitpunkt sich das Video gerade befindet. Die SceneView stellt mehrere Zeitpunkte im Detail dar. An dem linken und rechten Rand werden Zeitstempel bezüglich der Startzeit, beziehungsweise Endzeit der entsprechenden Szene angezeigt. Wenn die aktuelle Position des Videos innerhalb der SceneView liegt, so wird auch diese Position durch einen Zeitstempel angezeigt. Damit der Benutzer auch bei der Interaktion mit der SceneView einen zeitlichen Kontext behält, wird außerdem ein Zeitstempel an der horizontalen Position des Mauszeigers angezeigt.

4.8 Filter

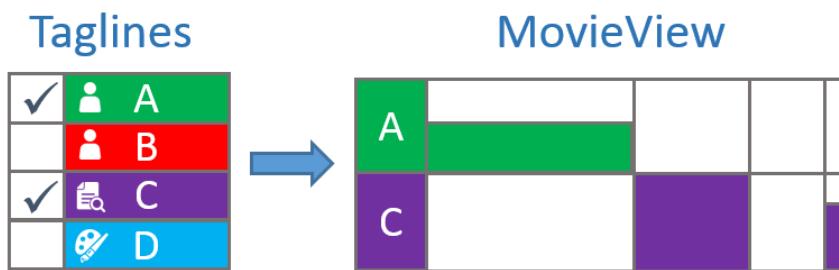


Abbildung 4.16: Ausgewählte Taglines in der Liste (links) werden in der MovieView angezeigt.

Standardweise werden in der MovieView die fünf Taglines mit den meisten Tags angezeigt. Das erlaubt es einen guten ersten Überblick über die Hauptcharaktere oder andere wichtige Facetten eines Films zubekommen. Liegt der Fokus einer Analyse aber beispielsweise auf der Beziehung zweier bestimmter Charaktere, so ist die Standardansicht nicht ausreichend. Zwei Taglines, die in der MovieView nicht direkt benachbart sind, sind außerdem schwieriger zu vergleichen. Aus diesen Gründen können Taglines gefiltert werden. Dafür werden die Taglines, für die sich ein Nutzer interessiert, wie in Abbildung 4.16 zu sehen, ausgewählt, woraufhin die MovieView genau diese Taglines anzeigt. Die Höhe der MovieView bleibt dabei konstant, sodass bei einer kleineren Anzahl an ausgewählten Taglines, die einzelnen Tagline eine größere Höhe haben. Das erlaubt es die Höhen der einzelnen Balken besser zu erkennen und zu vergleichen. Auch die Taglines aller ausgewählten SceneViews werden aktualisiert und angezeigt.

4.9 Kombination von Taglines

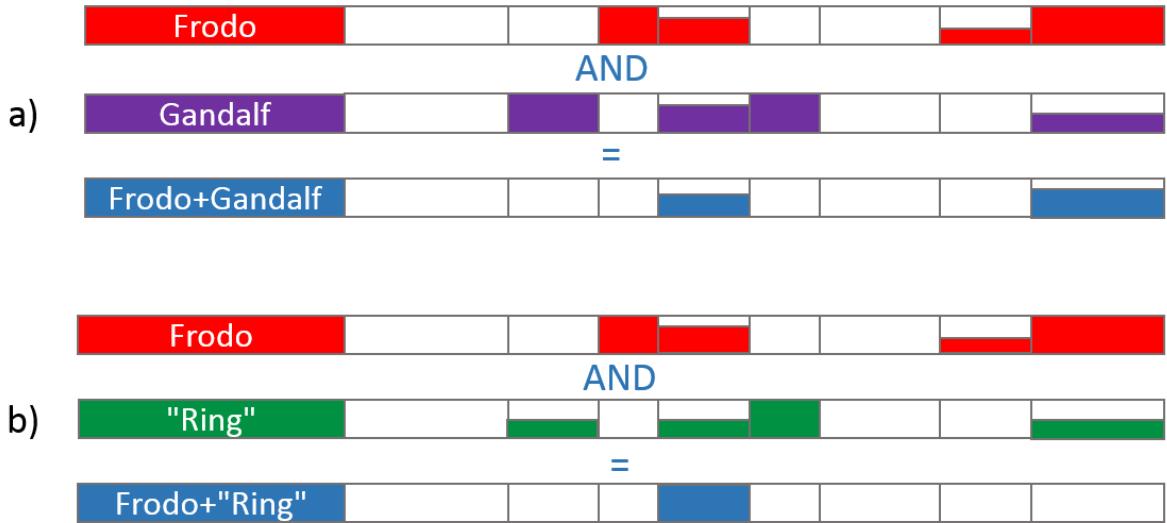


Abbildung 4.17: Die Kombination von Taglines. a) Kombination von Filmcharakteren. b) Kombination von Filmcharakter und Suchbegriff.

Um die Beziehung von Charakteren zu analysieren, ist es vorteilhaft die Szenen zu sehen, in denen mehrere Charaktere gemeinsam auftreten. Eine weiterer Anwendungsfall ist es, alle Filmstellen zu sehen, in denen ein Charakter einen bestimmten Suchbegriff sagt. Deshalb können Taglines kombiniert werden. Bei einer Kombination von Taglines, die zu Filmcharakteren gehören (siehe a) in Abbildung 4.17), wird ein Schnitt der Tags auf Szenenebene durchgeführt. Die kombinierte Tagline enthält genau bei den Szenen Tags, in denen alle Taglines mindestens einen Tag haben. Dadurch können in der kombinierten Tagline alle Szenen erkannt werden, in denen alle Filmcharaktere vorkommen. Bei einer Kombination von Taglines, bei der mindestens eine Tagline zu einem Suchbegriff gehört (siehe b) in Abbildung 4.17), zu sehen ist, entsteht eine kombinierte Tagline, bei der jeder Tag diesen Suchbegriff enthält. Dadurch können schnell Szenen und Filmstellen gefunden werden, in denen ein Filmcharakter etwas bestimmtes sagt.

4.10 Aufbau der Oberfläche

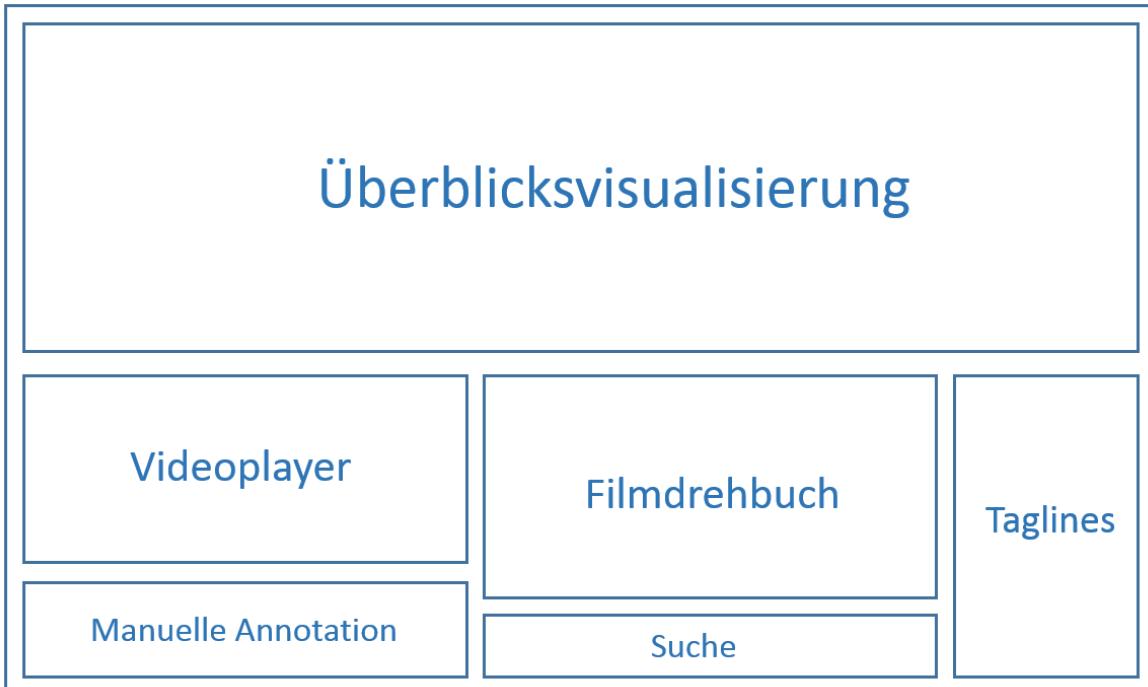


Abbildung 4.18: Der schematische Aufbau der Oberfläche. Der Überblicksvisualisierung wird der größte Platz zugeordnet. Darunter werden der Videoplayer und das Filmdrehbuch-Panel platziert. Weitere Elemente werden diesen sinnvoll platziert.

In den vorhergehenden Abschnitten wurden die Konzepte für die Überblicksvisualisierung und andere wichtige Bestandteile des Systems vorgestellt. Im Folgenden wird der Aufbau und die Struktur der einzelnen Elemente in einer graphischen Oberfläche beschrieben. Abbildung 4.18 zeigt den schematischen Aufbau der Oberfläche. Die obere Hälfte der Fläche wird von der Überblicksvisualisierung eingenommen, da diese den Hauptfokus des Systems darstellt. Sie breitet sich über die volle Breite der gegebenen Fläche aus, damit auch kurze Szenen in der MovieView noch ausreichend breit sind und erkannt werden können. Unterhalb der Überblicksvisualisierung werden der Videoplayer und das Filmdrehbuch-Panel nebeneinander platziert. Das begründet sich durch die enge Verknüpfung dieser beiden Komponenten durch die Synchronisation und Interaktion (siehe Abschnitt 4.4). Da die Suche eng mit dem Text des Filmdrehbuchs verknüpft ist, wird diese unterhalb des Filmdrehbuch-Panels platziert. Die Liste der Taglines, die es ermöglicht Taglines zu filtern und Taglines zu kombinieren wird in der unteren, rechten Eck platziert, da sie eine möglichst große Höhe benötigt, um so viele Taglines wie möglich anzuzeigen und unnötiges scrollen zu verhindern.

5 Implementierung

In diesem Kapitel werden das entwickelte System und die wichtigsten Aspekte der Implementierung des Systems beschrieben. Zuerst wird ein Überblick über den entwickelten Prototyp gegeben. Danach wird die Architektur des Systems erläutert, bevor auf die wichtigsten Module näher eingegangen wird. Es werden außerdem technische Designentscheidungen erklärt und begründet.

5.1 Systemübersicht

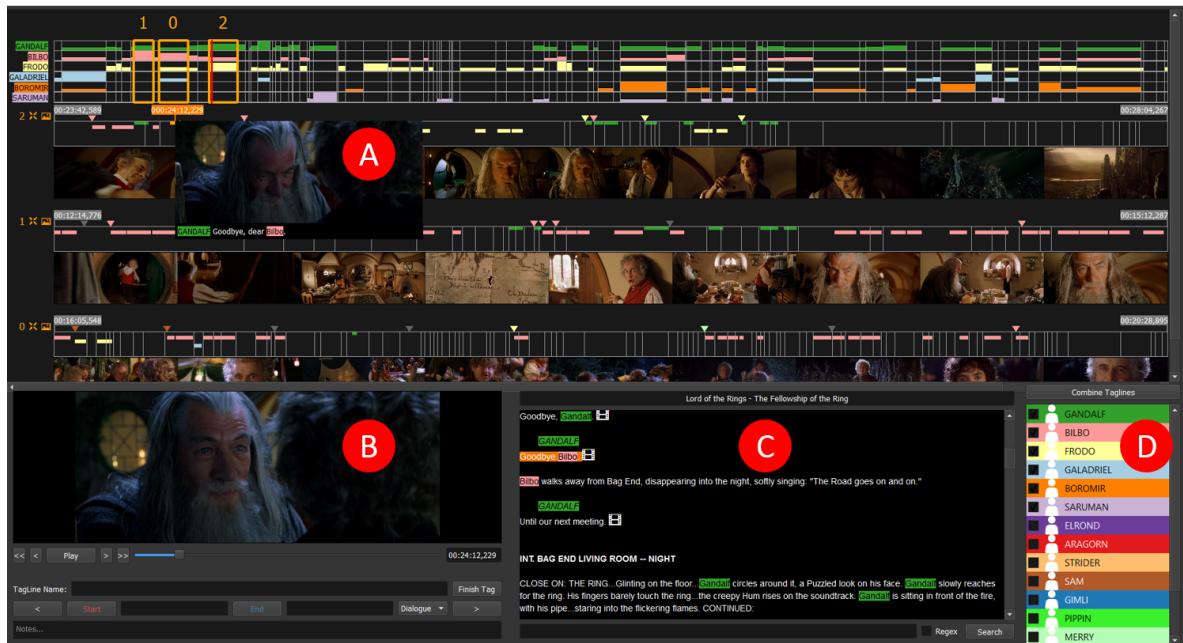


Abbildung 5.1: Ein Überblick über das entwickelte System. Als Datengrundlage dient dabei der Film *Der Herr der Ringe: Die Gefährten*.

Abbildung 5.1 zeigt das entwickelte System. Die wichtigsten Bestandteile wurden dabei mit Buchstaben markiert. Die Überblicksvisualisierung (A) bildet den Hauptteil des Systems. Hier ist am oberen Rand die MovieView, sowie die SceneViews der drei ausgewählten Szenen zu sehen. In der unteren Hälfte befindet sich der Videoplayer (B) und das Filmdrehbuch-Panel (C). Die Liste der vorhandenen Taglines (D) befindet sich in der rechten, unteren Ecke.

5.2 Farbgebung

Die Farbgebung spielt eine wichtige Rolle für die Verständlichkeit, Übersichtlichkeit und Erlernbarkeit einer Visualisierung [LFK⁺13]. Bei der Entwicklung der Überblicksvisualisierung wurde daher ein besonderer Fokus auf die Farbgebung gelegt. Wichtig ist bei der Farbgebung vor allem Konsistenz. Visuelle Elemente, welche die gleiche Entität repräsentieren sollten stets die gleiche Farbe haben. Auch sollten die gewählten Farben möglichst distinkt und leicht unterscheidbar sein.

Im Folgenden wird beschrieben wie diese beiden Aspekte umgesetzt wurden.

5.2.1 Konsistenz

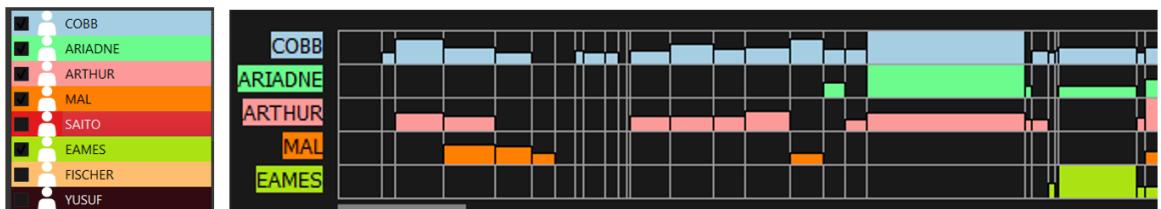


Abbildung 5.2: Jeder Tagline wird eine Farbe zugeordnet. Elemente, die zu der Tagline gehören, werden in der selben Farbe dargestellt.

Jeder Tagline wird bei seiner Erzeugung eine eindeutige Farbe zugeordnet. In dieser Farbe werden in der MovieView und in der SceneView auch die einzelnen Tags gefärbt. Dadurch können sie leichter ihrer zugehörigen Tagline zugeordnet werden. Die Label am linken Rand der MovieView und der maximierten SceneView, die den Namen einer Tagline anzeigen haben ebenfalls diese Farbe als Hintergrundfarbe. In der Listenansicht aller Taglines wird die Farbe als Hintergrundfarbe der einzelnen Listenelemente verwendet.

Für Taglines von Filmcharakteren geht diese Farbkonsistenz noch weiter. Wenn der Text einer Aktion des Filmdrehbuchs den Namen eines Filmcharakters enthält, so wird das entsprechende Dreieck in der SceneView in der Farbe der Tagline dieses Charakters gezeichnet. Enthält der Text keinen Filmcharakter, so wird ein neutraler Grauton verwendet. Enthält der Text mehrere Filmcharaktere so wird die Farbe des Charakters genommen, der zuerst erwähnt wird.

Namen von Filmcharakteren werden in allen Texten und Tooltips für eine schnelle und leichte Erkennbarkeit farblich markiert. Dies geschieht in den Tooltips von Aktionen, Tags, Notizen und auch im eigentlichen Text des Filmdrehbuchs im Filmdrehbuch-Panel.

Allen gliedernden und strukturellen Elementen der Visualisierung wird ein heller Grauton zugeordnet. Das betrifft die vertikalen Trennlinien, welche die Szenen- und Shotgrenzen in der SceneView bzw. MovieView repräsentieren, aber auch die horizontalen Trennlinien zwischen den Taglines. Der Grund für eine solche neutrale und unauffällige Farbe ist, damit diese Elemente nicht vom eigentlichen Fokus, den Taglines, ablenken. Gleichzeitig heben sich diese Elemente aber gut vom dunklen Hintergrund ab und sind leicht erkennbar.

Auch die Interaktion kann durch eine sinnvolle und konsistente Farbgebung verbessert werden. Dafür wird die Farbe Orange innerhalb der Visualisierung und des gesamten Systems für das Konzept "Selektion" verwendet. Immer wenn der Benutzer ein visuelles Element der Visualisierung auswählt, wird dies mit der Farbe Orange signalisiert.

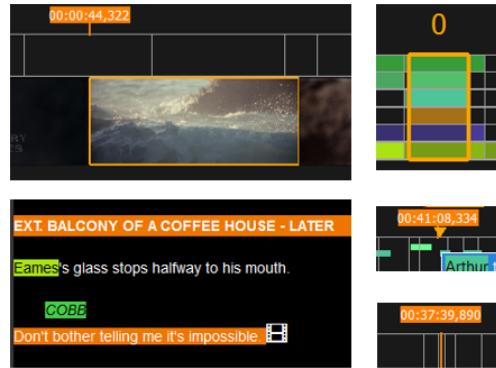


Abbildung 5.3: Die aktuelle Selektion des Benutzers wird durch die Farbe Orange repräsentiert.

Fährt der Benutzer mit dem Mauszeiger über die Rechtecke der Szenen in der MovieView, wird ihr Rand orange gefärbt. Beim Klicken wird der Rand breiter und bleibt permanent orange. Gleiches gilt für den Tag einer Tagline. Fährt der Benutzer mit dem Mauszeiger über die SceneView, so wird der Zeitpunkt, den die horizontale Position der Maus repräsentiert, in einem orangenen Zeitstempel am oberen Rand der SceneView dargestellt. Wenn der Benutzer über einen Frame des Storyboards fährt, wird der selektierte Frame orange umrandet und der entsprechende Zeitstempel erscheint wieder am oberen Rand der SceneView. Zusätzlich werden die relevanten Textstellen (Szenenbeschreibungen, Dialoge) im Filmdrehbuch-Panel orange markiert wenn die entsprechenden Elemente der SceneView ausgewählt werden. Diese Farbkonsistenz erleichtert es dem Benutzer einen guten Überblick über die Visualisierung zu behalten, da zu jedem Zeitpunkt klar ist welche Elemente der Visualisierung ausgewählt oder aktiv sind.

5.2.2 Distinkтивität

Die Farben der einzelnen Taglines sollten möglichst distinkt sein, damit sie in der MovieView und vor allem in der minimierten SceneView leicht zu unterscheiden sind. Sonst kann es leicht zu Verwirrungen und Missverständnissen seitens des Benutzers kommen, und die Benutzerfreundlichkeit und Verständlichkeit der Visualisierung sinkt. Besonders wichtig ist, dass die Farben der wichtigsten Filmcharaktere sich möglichst stark unterscheiden, da diese am häufigsten miteinander verglichen werden.

5 Implementierung

Die primäre Farbpalette wurde von ColorBrewer¹ generiert. Es wurde dabei eine Farbpalette mit zwölf verschiedenen Farben gewählt. Diese Farben haben die Eigenschaft, dass sie sich gut von einander unterscheiden.

Für die zwölf wichtigsten Taglines, in der Regel sind das die am häufigsten vorkommenden Filmcharaktere, werden die zuvor generierten Farben zugeordnet. Da es aber möglich ist, dass mehr als zwölf Taglines gleichzeitig im System vorhanden sind, werden für alle weiteren Taglines zufällige Farben erzeugt, die sich von den zwölf Hauptfarben unterscheiden. Dies erscheint als ein sinnvoller Kompromiss, da es schwierig ist eine diskidente Farbpalette für eine hohe Anzahl an Taglines zu finden.

Ein Problem, das bei der Entwicklung der Visualisierung im Bezug auf die Farbgebung aufgetreten ist, dass manche Taglines sehr helle Farben zugeordnet wurden. Da nun aber, wie zuvor beschrieben, Labels und Text mit dieser Farbe als Hintergrundfarbe gefüllt werden, und die Standardtextfarbe des Systems weiß ist, hatten diese Texte wegen des geringen Kontrastes eine schlechte Lesbarkeit.

Aus diesem Grund wird die Textfarbe an die Hintergrundfarbe angepasst. Dafür wird zuerst die relative Luminanz $Y = 0.2126R + 0.7152G + 0.0722B$ [CRCV08] der Farbe berechnet. Die relative Luminanz ist ein Wert, der die Helligkeit einer Farbe angibt. Je höher dieser Wert, desto heller erscheint die Farbe für das menschliche Auge. Dabei haben die verschiedenen Farbkanäle eine verschiedene Gewichtung für die Luminanz. Der Grünanteil einer Farbe hat dabei den größten Einfluss auf die Helligkeit, der Blauanteil den kleinsten. Ist $Y > 0.5$, so wird die Farbe als ausreichend hell angesehen und es wird eine schwarze Textfarbe verwendet. Gilt dagegen $Y \leq 0.5$, so wird Weiß als Textfarbe gewählt.

¹<http://colorbrewer2.org/>

5.3 Architektur

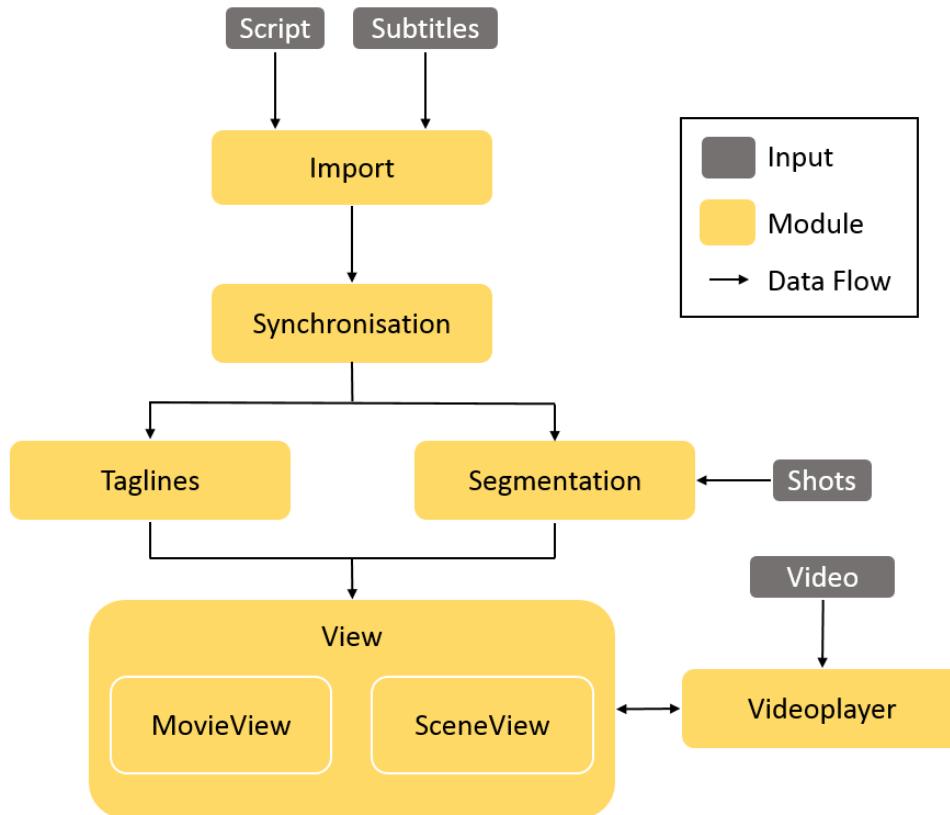


Abbildung 5.4: Die Architektur des vorgestellten Systems.

Die Architektur des Systems ist modular aufgebaut.

Das **Import**-Modul ist für das Einlesen und für die Konvertierung der Untertitel und des Filmdrehbuchs in Modellklassen (siehe Abschnitt 5.4) verantwortlich. In dieser Form können nun die Untertitel und Dialoge des Filmes von dem Modul **Synchronisation** verarbeitet werden. Dieses Modul implementiert die in Abschnitt 2.3 beschriebenen Ähnlichkeitsmaße und verknüpft ähnliche Dialoge und Untertitel. Eine nähere Beschreibung dieses Moduls folgt in Abschnitt 5.7. Das Modul **Segmentation** benutzt die Ergebnisse der Synchronisation und die Shottransitionen als Eingabe und führt eine hierarchische Segmentation des Eingabevideos in Szenen und Shots durch. Das Modul **Taglines** generiert die Taglines der automatisch erkannten Filmcharaktere aus den Ergebnissen der Synchronisation. Wenn der Benutzer nach einem Begriff sucht, manuelle Annotationen erstellt oder vorhandene Taglines kombiniert, so generiert dieses Modul die entsprechenden Taglines.

Nun stehen alle notwendigen Daten bereit, um die eigentliche Visualisierung aufzubauen. Dafür ist das Modul **View** verantwortlich. Dieses benutzt die aus den vorherigen Modulen erzeugten Daten und bildet diese auf ihre visuellen Repräsentationen ab. Auch ist die View für jegliche Interaktionen des

Benutzers mit der Visualisierung zuständig. Die View gliedert sich weiter unter in die einzelnen Be standteile der Überblicksvisualisierung, wesentlich sind hier die **MovieView** und die **SceneView**. Die View steht durch die Interaktion in gegenseitiger Beziehung mit dem **Videoplayer**. Der Videoplayer nimmt das Video als Eingabe und ist für die Wiedergabe und Interaktion des Videos verantwortlich.

5.4 Datenmodell

Im Folgenden wird das Datenmodell des Systems beschrieben.

Im System müssen verschiedene Arten von Daten verwaltet werden. Dazu zählen unter anderem die Untertitel, das Filmdrehbuch, und die Videos. Diese Daten sind auch untereinander verknüpft. Diese Relationen müssen ebenfalls modelliert werden.

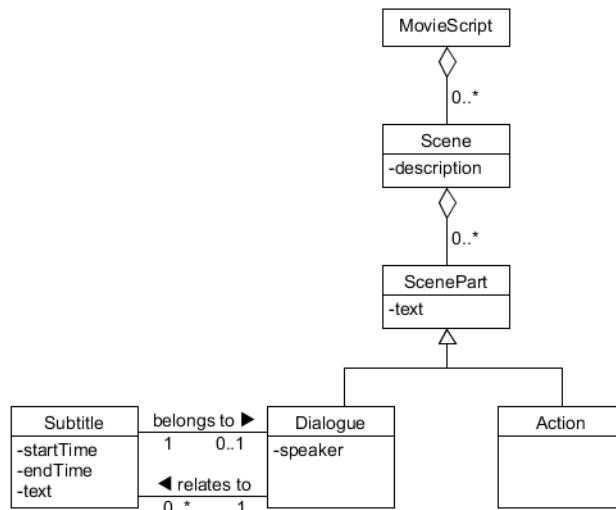


Abbildung 5.5: Dieses Klassendiagramm modelliert Untertitel, Filmdrehbücher und ihre Relationen.

Abbildung 5.5 zeigt das UML-Klassendiagramm des Datenmodells. Diese Darstellung zeigt eine hierarchische Struktur der einzelnen Klassen. Ein Filmdrehbuch besteht aus Szenen. Diese Szenen wiederum haben einzelne Szenenteile. Diese Szenenteile können, wie in 5.5.2 beschrieben, Dialoge oder Aktionen sein.

Dieser Sachverhalt wird im Klassendiagramm wie folgt modelliert. Ein Filmdrehbuch wird durch die Klasse *MovieScript* repräsentiert. Eine Szene wird durch die *Scene*-Klasse realisiert. Zwischen diesen Klassen besteht eine Komposition.

Um den Aspekt zu modellieren, dass eine Szene aus verschiedenen, aufeinanderfolgenden Teilen besteht, wird das Konzept der Vererbung genutzt. Jeder Szenenteil wird durch ein *ScenePart* dargestellt. *ScenePart* wiederum ist die Vaterklasse von *Dialogue* und *Action*. Diese beiden letzten Klassen stellen

die möglichen Arten von Szenenteilen dar. Beide haben gemeinsam, dass sie einen Text besitzen. Ein Dialogue hat zusätzlich einen *Speaker*, da jeder Dialog einen Sprecher hat.

Ein zentraler Aspekt dieser Arbeit ist die Verknüpfung von Bild- und Textinhalten von Filmen. Dafür müssen Untertitel, die über eine zeitliche Komponente verfügen, mit den Dialogen des Filmdrehbuchs verknüpft werden. Untertitel werden durch die *Subtitle*-Klasse realisiert, und haben eine bidirektionale Assoziation mit Dialogue. Ein Dialogue kann beliebig viele, aber auch keinen, verwandten Subtitle haben. Dies liegt daran, dass ein Dialog möglicherweise keine entsprechende Stelle in den Untertiteln hat, da das Filmdrehbuch nicht aktuell ist oder bis zum Drehen des Films geändert wurde. Außerdem kann ein Dialog mehrere verwandte Untertitel haben, da Dialoge in der Regel ganze Sätze oder Satzreihen sind. Ein Dialog kann also auf mehrere Untertitel gemappt werden. Untertitel hingegen haben eine kürzere Länge, und werden auf einen oder keinen Dialog gemappt.

5.5 Import

Um die Visualisierung aufzubauen müssen erst die Eingabedaten analysiert werden. Als Eingabe benötigt das System einen Film, die dazugehörigen Untertitel und einen Filmdrehbuch.

5.5.1 Untertitel

Die Untertitel liegen dabei im SRT (SubRip Text)-Format vor. SRT ist eine der einfachsten Arten Untertitel zu kodieren [Mat].

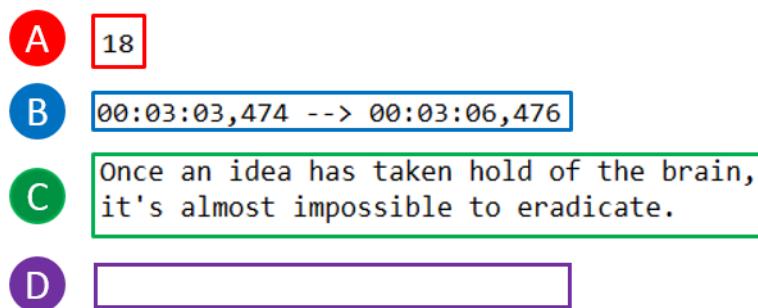


Abbildung 5.6: Die Struktur einer SRT-Datei.

SRT-Dateien haben einen simplen Aufbau. Jeder Untertitel besteht dabei aus vier Teilen (siehe Abbildung 5.6):

- A** Eine Zahl die den Index des aktuellen Untertitels im gesamten Dokument angibt, beginnend bei 1.

B Angabe zu welchem Zeitpunkt der Untertitel eingeblendet werden muss, und wann dieser wieder ausgeblendet werden muss. Diese Angabe ist stets wie folgt formatiert:

Stunde:Minute:Sekunde,Millisekunde -> Stunde:Minute:Sekunde,Millisekunde

Der Teil vor dem Pfeil definiert die Startzeit des Untertitels, und der Teil nach dem Pfeil die Endzeit.

C Der eigentliche Text des Untertitels. Dieser kann sich auch über mehrere Zeilen erstrecken.

D Eine Leerzeile, die das Ende des aktuellen Untertitels signalisiert.

Mit Hilfe von regulären Ausdrücken lassen sich die Zeitangaben leicht verarbeiten. Das System konvertiert die Start- und Endzeit dann zu Millisekunden, da alle Berechnungen mit Zeitangaben im System auf Millisekundenebene stattfinden. Dies ist vor allem dadurch begründet, dass der in 5.9 beschriebene QMediaPlayer, es erlaubt zu beliebigen Zeitpunkten eines Videos in Millisekunden zu springen.

5.5.2 Filmdrehbuch

Im Gegensatz zu Untertiteln gibt es für Filmdrehbücher keine standardisierten Dateiformate. Filmdrehbücher können als PDF-Dateien oder normale Textdateien vorliegen. Auch für die Formatierung eines Filmdrehbuchs existieren keine klaren Regeln.

A
INT. WORKSHOP - CONTINUOUS

B
Ariadne thinks this over.

ARIADNE C
That's not an issue for me.

ARTHUR D
Why not?

ARIADNE
Arthur, maybe you can't see what's going on, maybe you don't want to. But Cobb's got problems he's tried to bury down there. I'm not going to open my mind to someone like that.

Ariadne gets to her feet. Walks away.

COBB (O.S.) E
She'll be back.

Abbildung 5.7: Auszug vom Filmdrehbuch von *Inception*

Dennoch haben die meisten Filmdrehbücher eine ähnliche Struktur, was eine automatisierte Verarbeitung ermöglicht. Abbildung 5.7 zeigt beispielhaft einen Ausschnitt eines Drehbuchs. Die wichtigsten Bestandteile werden dabei durch Buchstaben gekennzeichnet. Die Bezeichnung der einzelnen strukturellen Elemente ist dabei an [The] angelehnt.

- A Szenenbeschreibung** Die Szenenbeschreibung steht zu Beginn jeder Szene und gibt Auskunft über den Ort und die Zeit der Szene. Die Abkürzung *Int.* steht dabei für *interior* und zeigt an, dass diese Szene in einem geschlossenen Raum befindet. Daneben existiert noch *Ext.* für *exterior*, welches bei Szenen verwendet wird, die draußen stattfinden.
- B Aktion**
Die Aktion enthält Beschreibungen zu den Ereignissen eines Films. Meistens wird dabei beschrieben wie Charaktere eine Handlung durchführen. Die Aktion wird in der Regel im Präsens geschrieben.
- C Charaktername**
Der Charaktername gibt Auskunft über den Sprecher des darauffolgenden Dialogs. In der Regel bleibt dieser Name über das gesamte Filmdrehbuch konsistent.
- D Dialog**
Der Dialog gibt das an, was ein Charakter des Films sagt.
- E Extension**
Die Extension liefert Informationen darüber, wie die Stimme eines Charakters im Folgenden Dialog gehört wird.
Ein Charakter der einen Dialog sagt, während er nicht sichtbar ist, wird beispielsweise als *(O.S.)*, *off-screen* bezeichnet.

Die hier beschriebenen Teile des Drehbuchs werden beim Import erkannt und eingelesen. Als Quelle für Filmdrehbücher wurde IMSdb² benutzt. Hier liegen Drehbücher meistens als Textdatei vor. Aus diesem Grund unterstützt das System den Import von Drehbüchern im Textformat. Da die Formatierung der Drehbücher von Film zu Film unterschiedlich ist, gestaltet sich ein vollautomatischer Import als schwierig. Um die Implementierung zu erleichtern wurde deshalb ein semi-automatisches Verfahren benutzt. Hierbei müssen für den Import eines Drehbuchs zusätzlich manuell extrahierte Informationen zu der Formatierung angegeben werden. Das betrifft vor allem die Einrückung der verschiedenen Teile eines Drehbuchs.

5.6 Zeitliche Segmentierung

Die zeitliche Segmentierung des Eingabevideos erfolgt auf zwei Ebenen.

Zum einen wird eine Shotsegmentierung durch einen Algorithmus, basierend auf dem Prinzip von Bewegungsvektoren (siehe Abschnitt 2.1), durchgeführt. Dieser Algorithmus wurde nicht im Rahmen dieser Arbeit entwickelt, sondern wurde vom Institut für Visualisierung und Interaktive Systeme

²<http://www.imsdb.com/>

5 Implementierung

der Universität Stuttgart bereit gestellt. Zuerst wird das Eingabevideo dem Algorithmus übergeben, welcher eine Textdatei mit den Frames liefert, bei denen ein neuer Shot beginnt. Diese Textdatei wird dann bei dem Start des Systems eingelesen und daraus die Zeiten der Shottransitions berechnet.

Zum anderen erfolgt eine Segmentierung auf Szenenebene. Dabei wird das Video basierend auf den Ergebnissen der Synchronisation und den Filmdrehbüchern in Szenen unterteilt.

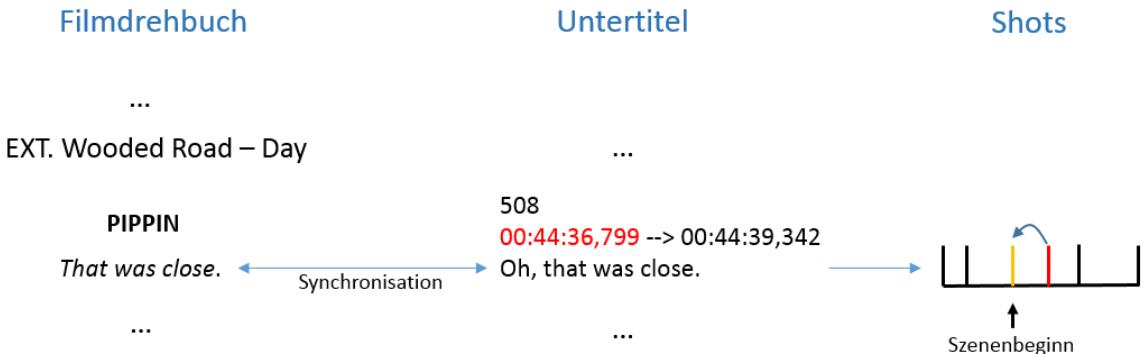


Abbildung 5.8: Schematische Darstellung der Szenensegmentierung.

Diese Szenensegmentierung funktioniert wie folgt: Für jede Szene s des Filmdrehbuchs wird der erste Dialog d bestimmt, für den der Synchronisationsalgorithmus einen passenden Untertitel u gefunden hat. Die Startzeit t_{start} von u ist nun logischerweise eine obere Schranke für die Startzeit von s , da s frühestens mit dem Beginn des ersten Dialogs anfangen kann. Da Szenentransitionen üblicherweise auch mit Shottransitions einhergehen, werden die Ergebnisse der Shotsegmentierung hier ebenfalls genutzt. Als Startzeitpunkt von s wird der Zeitpunkt der ersten Shottransition gewählt, die vor t_{start} liegt.

Wurden zu Beginn von s mehreren Dialogen keine Untertitel zugeordnet, so ist die berechnete Startzeit von s wesentlich weiter in der Zukunft als die eigentliche Startzeit und somit ungenau. Um diesen Effekt entgegenzuwirken könnte die Endzeit der s vorangehenden Szene s_{prev} für die Berechnung der Startzeit einbezogen werden. Hier könnte zum Beispiel das arithmetische Mittel der Endzeit von s_{prev} und der Startzeit des ersten verknüpften Untertitels von s berechnet werden, und dann als Startzeit festgelegt werden.

5.7 Synchronisation

In diesem Abschnitt wird die Synchronisation von den Untertiteln und den Dialogen eines Filmdrehbuchs behandelt.

Warum ist diese Synchronisation überhaupt notwendig? Bei einer Betrachtung von Filmdrehbüchern und dazugehörigen Filmen fällt schnell auf, dass genaue Wortlaute, die Reihenfolge von Worten oder ganzen Sätzen zwischen den beiden Medien variieren. Oft enthält das Drehbuch veraltete Teile die im

fertigen Film nicht mehr zu finden sind. Dennoch wollen wir die im Filmdrehbuch enthaltenen Metadaten für unsere Überblicksvisualisierung nutzen. Drehbücher enthalten zum einen eine semantische Segmentierung des Films in einzelne Szenen. Das erlaubt es Szenen zu analysieren und visualisieren. Zum anderen beinhalten sie die Information, welcher Filmcharakter welchen Teil des Dialogs sagt. Dies ist nützlich, da Untertitel im Allgemeinen keinerlei Information über den inhaltlichen Kontext hergeben, aber dafür genaue Zeitstempel der Dialoge enthalten. Das Filmdrehbuch ordnet also Dialoge zu Charakteren zu, und die Untertitel ordnen gesprochenen Text einem Zeitintervall zu.

Ein Untertitel sei definiert durch ein Text, einen Startzeitpunkt, und einen Endzeitpunkt. Ein Dialog sei definiert durch seinen Text, sowie den Namen des Sprechers. Gegeben sei eine Liste von Untertiteln U , und eine Liste von Dialogen D .

Die Synchronisation wird als folgendes Problem definiert:

Ordne jedem Element u_i aus U ein Element d_i aus D zu, deren Texte sich am meisten ähneln.

Das Ziel ist nun eine möglichst gute Zuordnung zu finden.

Ein Untertitel soll genau dann einem Dialog zugeordnet werden, wenn er eine hohe Ähnlichkeit zu diesem besitzt. Diese Ähnlichkeit soll zum einen auf semantischer Ebene sein. Zum anderen spielt auch die lexikalische Gleichheit eine wichtige Rolle. Sätze, die eine hohe Anzahl an gemeinsamen Wörtern haben, besitzen vermutlich auch eine hohe Ähnlichkeit.

Die bestmögliche Art der Ähnlichkeit ist die Gleichheit zweier Texte. Wir definieren dafür das Ähnlichkeitsmaß $\text{sim}_{\text{equal}}$:

$$\text{sim}_{\text{equal}}(t_1, t_2) = \begin{cases} 1 & \text{if } t_1 = t_2 \\ 0 & \text{else} \end{cases}$$

Diese Überprüfung wird zu Beginn des Synchronisationsalgorithmus durchgeführt, da die Prüfung auf Gleichheit die strikteste ist. Das bedeutet, dass die Gleichheit zweier Sätze impliziert, dass die beiden Sätze auch von jedem anderen (sinnvollen) Ähnlichkeitsmaß als ähnlich klassifiziert werden. Ist ein Satz ausreichend lang, d.h. nicht nur eine kurzer Ausruf oder ein nur ein Name eines Filmcharakters, so ist bei Gleichheit der zugeordnete Dialog mit hoher Wahrscheinlichkeit auch der entsprechende Dialog des Filmdrehbuchs.

Die nächste bestmögliche Art der Ähnlichkeit ist das vollständige Enthalten des Untertiteltextes im Dialogtext als Teilwort. Auch dafür definieren wir ein Ähnlichkeitsmaß:

$$\text{sim}_{\text{substr}}(t_1, t_2) = \begin{cases} 1 & \text{if } t_2 \text{ contains } t_1 \\ 0 & \text{else} \end{cases}$$

Ist eine Untertitel komplett in einem Dialog enthalten, also ein Substring eines Dialogs nach der Normalisierung, und ist der Untertitel lang genug, so ist auch hier vermutlich der korrekte Dialog zu dem Untertitel gefunden worden.

5 Implementierung

Diese beiden Ähnlichkeitsmaße haben den Vorteil, dass sie eine hohe Präzision haben. Die Zuordnung von Untertiteln und Dialogen, die gefunden wurden, sind meistens korrekt. Der Nachteil ist aber, dass viele eigentlich korrekte Zuordnungen nicht gefunden werden. Ist der Unterschied zwischen Untertitel und Dialog sehr gering, wenn beispielsweise nur die Reihenfolge zweier Wörter vertauscht ist, so werden die Ähnlichkeitsmaße diese beiden Elemente nicht einander zuordnen.

Aus diesem Grund müssen andere Techniken verwendet werden, die auch Sätze mit geringen lexikalischen Änderungen zuordnen können. In Abschnitt 2.3 wurden verschiedene Ähnlichkeitsmaße für den Vergleich von Sätzen vorgestellt. Aus diesen wurde das TF-IDF Ähnlichkeitsmaß $\text{sim}_{\text{tfidf}}$, der Jaccard Similarity Coefficient sim_{jfc} , und die abgewandelte Form des Jaccard Similarity Coefficient $\text{sim}_{\text{jfc}^*}$ implementiert.

Es wurde außerdem eine Kombination dieser Ähnlichkeitsmaße $\text{sim}_{\text{combined}}$ entwickelt, bei der die verschiedenen Maße hintereinander angewendet werden. Die Plausibilität dieser Idee begründet sich wie folgt: Werden erst die striktesten Ähnlichkeitsmaße $\text{sim}_{\text{equal}}$ und $\text{sim}_{\text{substr}}$ angewendet, so werden die Dialog-Untertitel Paare zugeordnet die mit hoher Wahrscheinlichkeit korrekt sind. Für die restlichen, nicht zugeordneten Paare können nun die anderen Ähnlichkeitsmaße verwendet werden, die auch bei kleineren Unterschieden von Sätzen noch eine Zuordnung durchführen. Dadurch wird erhofft, einen größeren Teil der korrekten Zuordnungen durchzuführen, als mit der Verwendung eines einzigen Ähnlichkeitsmaßes möglich wäre.

Da alle vorgestellten Verfahren eine Ähnlichkeit mit einem numerischen Wert im Bereich $[0, 1]$ bewerten, ist es einfach festzustellen wie sich die Ähnlichkeit von einem Untertitel zu mehreren Dialogen verhält. Der ähnlichste Dialog d_{best} für einen gegebenen Untertitel ist somit leicht zu bestimmen. Dafür wird die Ähnlichkeit von einem Untertitel u zu allen Dialogen d_i einer Liste D berechnet, und das d_i , für das der Ähnlichkeitswert des Ähnlichkeitsmaßes sim maximal ist, gewählt:

$$(5.1) \quad d_{\text{best}} = \arg \max_{d \in D} \text{sim}(u, d)$$

Zusätzlich wird ein Schwellenwert ϵ mit $0 \leq \epsilon \leq 1$ eingeführt. Nur wenn die maximale Ähnlichkeit für ein gegebenes Ähnlichkeitsmaß diesen Wert übersteigt wird für einen Untertitel ein Dialog zugeordnet. Die Wahl des Schwellenwerts hat Auswirkungen auf die Ergebnisse der Synchronisation. Ist der Wert von ϵ eher klein, so verhält sich das Ähnlichkeitsmaß “optimistisch”, das heißt es werden schon bei kleineren Übereinstimmungen Dialoge zu Untertiteln zugeordnet. Bei einem großen ϵ ist das Ähnlichkeitsmaß “streuend” und ordnet weniger Paare zu, dafür aber mit einer höheren Präzision.

Listing 5.1 Die Implementierung des JFC-Ähnlichkeitsmaßes

```

1 Dialogue jfc(Subtitle subtitle, QList<Dialogue> dialogueList) {
2     QStringList subtitleWords = subtitle.text().split(" ")
3     Dialogue maxDialogue;
4     double maxJFC = 0;
5
6     for(Dialogue dialogue : dialogueList) {
7         QStringList dialogueWords = dialogue.text().split(" ");
8
9         //Compute number of common and distinct words
10        int commonWordCount = dialogueWords
11            .toSet()
12            .intersect(subtitleWords.toSet())
13            .length();
14        int distinctWordCount = dialogueWords
15            .toSet()
16            .unite(subtitleWords.toSet())
17            .length();
18        double jfc = (double) commonWordCount / (double) distinctWordCount;
19        if(jfc > maxJFC) {
20            maxJFC = jfc;
21            maxDialogue = dialogue;
22        }
23    }
24
25    // If the maximal jfc is above our threshold a matching dialogue is found
26    if(maxJFC > 0.8) {
27        return maxDialogue;
28    } else {
29        return NULL;
30    }
31 }
```

Listing 5.1 zeigt, wie das JFC-Maß im System implementiert wurde. Dafür wird eine Funktion definiert, die einen Untertitel und eine Liste von Dialogen als Parameter nimmt, und den Dialog zurückgibt der dem Untertitel am ähnlichsten ist. Diese berechnet zuerst den JFC-Wert des Untertitels zu allen Dialogen (Zeile 6 - 23), wobei der Dialog mit dem maximalen JFC-Wert gespeichert wird. Ist der maximale Wert über einem bestimmten Schwellenwert (Zeile 26), so wird der entsprechende Dialog zurückgegeben. Entsprechend wurden auch das abgewandelte JFC-Maß und das TF-IDF-Maß implementiert, weshalb für diese Ähnlichkeitsmaße kein Beispielcode dargestellt wird.

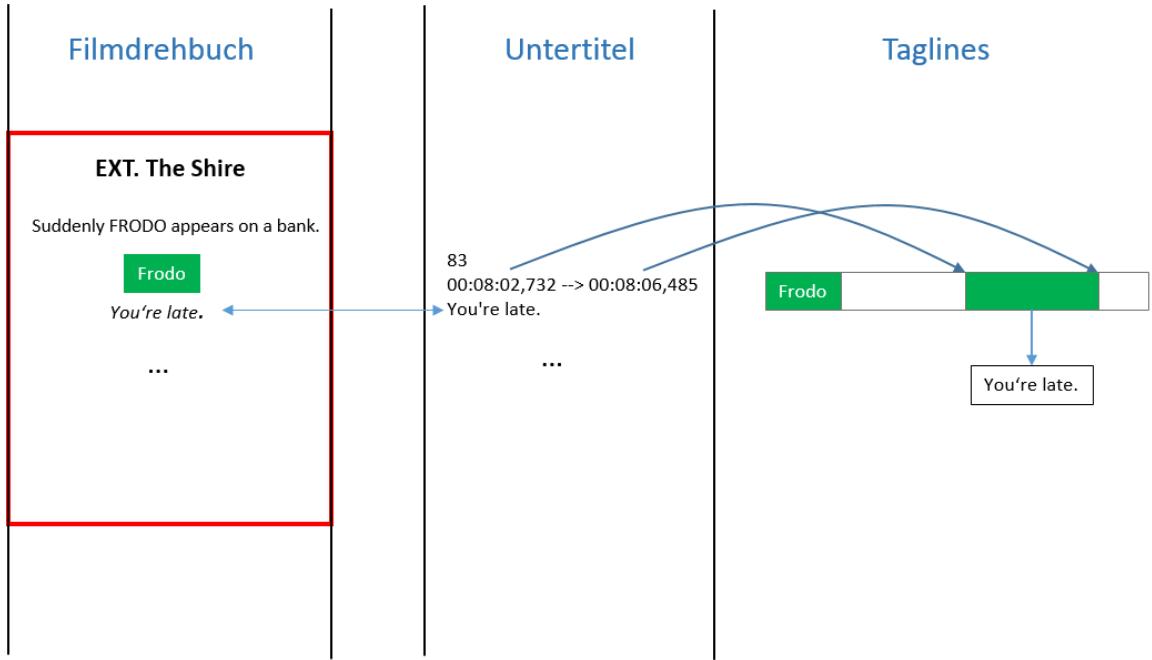


Abbildung 5.9: Die Synchronisation des Filmdrehbuchs und der Untertitel. Basierend auf der Synchronisation werden die Taglines der Filmcharaktere erstellt.

Abbildung 5.9 zeigt, wie das Filmdrehbuch und die Untertitel verknüpft werden. Nachdem die Synchronisation durchgeführt wurde, können die Ergebnisse der Synchronisation für die eigentliche Visualisierung verarbeitet werden. Dafür werden die Dialoge des Filmdrehbuchs betrachtet, für die ein entsprechender Untertitel existiert. Daraufhin werden die Sprecher der Dialoge geprüft, und für jeden Sprecher eine Tagline generiert. Jeder Dialog dieses Sprechers wird dann auf einen Tag abgebildet (siehe rechts in Abbildung 5.9). Der Text dieses Tags wird auf den Untertitel des entsprechenden Dialogs gesetzt.

5.7.1 Normalisierung

Eine nähere Betrachtung von Filmdrehbüchern und Untertiteln zeigt, dass oft nur kleine Formulierungen oder einzelne Worte selbige unterscheiden. So wird in einem Untertitel "that is" verwendet, während im Drehbuch "that's" verwendet wird oder die Zeichensetzung der Untertitel unterscheidet sich minimal von der Zeichensetzung im Drehbuch. Aus diesem Grund wird eine Normalisierung der Untertitel und Dialoge vorgenommen, bevor der eigentliche Synchronisationsalgorithmus ausgeführt wird.

Zum einen werden die gängigsten Kontraktionen auf ihre ursprüngliche Form gebracht. Außerdem werden Satzzeichen wie Fragezeichen, Ausrufezeichen und Kommata entfernt. Auch die Groß- und Kleinschreibung wird vereinheitlicht und sämtliche Texte werden in Kleinbuchstaben konvertiert. Dieser Prozess der Normalisierung senkt die Zahl der False Negatives, also der fälschlicherweise nicht

Listing 5.2 Eine verkürzte und vereinfachte Darstellung des Normalisierungsalgorithmus

```

1 void normalize(QString text) {
2     QRegExp punctuation("[\\?\\.,:-']");
3     QList<QPair<QString, QString>> contractions;
4     contractions << QPair("that's" , "that is");
5     contractions << QPair("they're", "they are");
6
7     QString normalizedText = text.toLower();
8
9     // Replace all contractions by their orginal form
10    for(QPair contraction : contractions) {
11        normalizedText = normalizedText
12            .replace(contraction.first, contraction.second);
13    }
14
15    // Remove all unnecessary punctuation and whitespace
16    normalizedText = text.
17        .replace(punctuation, "")
18        .trimmed();
19
20    return normalizedText;
21 }
```

zugeordneten Untertitel. Gleichzeitig steigt dadurch aber auch die Anzahl von False Positives, da zuvor nicht äquivalente Sätze sich nun stärker ähneln können.

Listing 5.2 zeigt einen Ausschnitt des Normalisierungsalgorithmus. Die Normalisierung wird mit Hilfe von regulären Ausdrücken (Zeile umgesetzt. Der Algorithmus spielt für die Synchronisation eine entscheidende Rolle, da sonst bereits kleinste Inkonsistenzen bei der Zeichensetzung oder Benutzung von Kontraktionen zwischen den Untertiteln und Dialogen zu einer Reduktion der Trefferquote führen würden.

5.8 Evaluation der Synchronisation

In diesem Abschnitt sollen die in 5.7 vorgestellten Synchronisationsalgorithmen evaluiert werden. Alle Ansätze haben gemeinsam, dass sie jedem Untertitel, wenn möglich, den ähnlichsten Dialog zuordnen. Diese Zuordnung muss für die Evaluation bewertbar sein.

Eine Zuordnung kann durch folgende Maße bewertet werden [AHS08]:

- Trefferquote
- Genauigkeit

Ein Untertitel, der zum korrekten Dialog zugeordnet wird, *True Positive* bezeichnet. Ein Untertitel, der zu einem falschen Dialog zugeordnet wird, wird als *False Positive* bezeichnet. Ein Untertitel, der zu keinem Dialog gehört, und korrekt erweise keinem Dialog zugeordnet wird, ist ein *True Negative*.

5 Implementierung

Zuletzt wird ein Untertitel, der zu keinem Dialog zugeordnet wird, jedoch eigentlich zu einem Dialog gehört, als *False Negative* bezeichnet.

Die Trefferquote T ist dabei das Verhältnis von korrekt klassifizierten Untertiteln zu allen Untertiteln die einen zugeordneten Dialog besitzen.

$$(5.2) \quad T = \frac{\text{Anzahl True Positives}}{\text{Anzahl True Positives} + \text{Anzahl False Negatives}}$$

Die Genauigkeit G ist das Verhältnis von korrekt klassifizierten Untertiteln zu allen zugeordneten Untertiteln.

$$(5.3) \quad G = \frac{\text{Anzahl True Positives}}{\text{Anzahl True Positives} + \text{Anzahl False Positives}}$$

Um die Trefferquote und Genauigkeit für einen Algorithmus zu bestimmen, müssen die korrekten Ergebnisse vorliegen. In diesem Fall bedeutet das, dass die Untertitel manuell mit den Dialogen verknüpft werden müssen. Die genaue Methodik für die Evaluation ist wie folgt: Für zwei verschiedene Filme (*Inception*, *Herr der Ringe - Die Gefährten*) wurden jeweils 300 Untertitel untersucht und manuell zu den entsprechenden Dialogen im Filmdrehbuch zugeordnet. Dabei wurden jeweils 100 Untertitel zu Beginn, in der Mitte, und am Ende des Filmes für die Evaluation einbezogen. Untertitel, die offensichtlich keine wörtliche Rede enthalten, wurden dabei übersprungen.

Ähnlichkeitsmaß	Trefferquote	Genauigkeit
sim _{equal}	0.14	1
sim _{substr}	0.32	0.9
sim _{jfc} $\epsilon = 0.5$	0.33	0.89
sim _{jfc} $\epsilon = 0.7$	0.22	0.89
sim _{jfc} $\epsilon = 0.9$	0.15	1
sim _{jfc*} $\epsilon = 0.5$	0.57	0.16
sim _{jfc*} $\epsilon = 0.7$	0.56	0.5
sim _{jfc*} $\epsilon = 0.9$	0.3	0.64
sim _{tfidf} $\epsilon = 0.5$	0.56	0.69
sim _{tfidf} $\epsilon = 0.7$	0.38	0.79
sim _{tfidf} $\epsilon = 0.9$	0.24	0.91
sim _{combined}	0.72	0.71

Tabelle 5.1: Die verschiedenen Ähnlichkeitsmaße und ihre Trefferquoten und Genauigkeiten.

Das Ähnlichkeitsmaß sim_{equal} hat wie erwartet eine sehr hohe Präzision. In diesem Fall ist sie mit 1 sogar maximal. Das heißt, dass bei der Prüfung auf Gleichheit alle Untertitel korrekt zugeordnet werden. Leider hat es aber mit 0.14 eine sehr niedrige Trefferquote, da in den meisten Fällen die Untertitel und Dialoge nicht komplett gleich sind.

Das Ähnlichkeitsmaß $\text{sim}_{\text{substr}}$ hat im Vergleich zu $\text{sim}_{\text{equal}}$ eine etwas höhere Trefferquote, aber besitzt dafür auch eine verringerte Präzision. Bei der Überprüfung auf Substrings werden also False Positives gefunden.

Bei dem Ähnlichkeitsmaß sim_{jfc} wurden verschiedene Schwellenwerte ϵ evaluiert. Für $\epsilon = 0.5$ ist die Trefferquote mit 0.33 im unteren Bereich, die Präzision dagegen mit 0.89 im höheren Bereich. Mit steigenden ϵ sinkt die Trefferquote entsprechend, während die Präzision auf 1 ansteigt. Das abgewandelte Ähnlichkeitsmaß $\text{sim}_{\text{jfc}}^*$ hat im Vergleich zu der ursprünglichen Version eine höhere Trefferquote. Das ist vermutlich dadurch begründet, dass bei diesem Maß die Anzahl der gemeinsamen Worte mit der Anzahl der Worte des ersten Satzes normalisiert wird, und nicht mit der Anzahl der Worte beider Sätze wie bei sim_{jfc} . Dementsprechend ist auch die Genauigkeit im Vergleich geringer. Für $\epsilon = 0.5$ hat dieses Maß die höchste Trefferquote aller bisher diskutierten Ähnlichkeitsmaße. Das Ähnlichkeitsmaß $\text{sim}_{\text{tfidf}}$ hat Trefferquoten die im selben Bereich wie $\text{sim}_{\text{jfc}}^*$ liegen. Gleichzeitig sind die Genauigkeiten hier wesentlich höher. Für $\epsilon = 0.5$ weißt dieses Maß mit 0.91 sogar eine der höchsten Genauigkeiten aller Verfahren auf.

Besonders interessant ist das letzte Ähnlichkeitsmaß $\text{sim}_{\text{combined}}$. Mit 0.72 hat dieses Maß die höchste Trefferquote. Durch eine Kombination von mehreren Ähnlichkeitsmaßen wird eine Trefferquote erreicht, welche die Trefferquote der einzelnen Ähnlichkeitsmaße übertrifft. Auch die Genauigkeit liegt mit 0.71 in einem akzeptablen Bereich. Somit stellt dieses Ähnlichkeitsmaß einen guten Kompromiss aus Trefferquote und Genauigkeit dar. Aus diesem Grund wird dieses Maß für die Synchronisation im System verwendet.

5.9 Technologien

Das System wurde mit Qt 5.4 entwickelt. Qt³ ist ein plattformunabhängiges Framework für die Entwicklung von Anwendungssoftware. Das Hauptanwendungsgebiet ist die Entwicklung von grafischen Oberflächen. Qt ist in der Programmiersprache C++ geschrieben und erweitert die Sprache durch eine Vielzahl von eigenen Klassen. Ein Besonderheit ist hier das Meta-Object System, welches die Kommunikation von Objekten mittels Slots und Signals erlaubt, und damit das Event Handling erleichtert.

Im Hinblick auf das in dieser Arbeit vorgestellte System, bietet Qt mehrere Vorteile. Die Wiedergabe von Videos gestaltet sich mittels des Qt Multimedia Moduls sehr einfach. Dies ist vorteilhaft da das Laden, Abspielen und Interagieren mit Videos ein wichtiger Aspekt dieser Arbeit ist. Konkret wurde hier der *QMediaPlayer* benutzt, der es ermöglicht ein Video zu laden und abzuspielen. Über das Slot- und Signalsystem ist es möglich zu einer bestimmten Millisekunde eines Videos zu springen. Dies wird vor allem benötigt um die Verknüpfung zwischen SceneView und Video herzustellen. Gleichzeitig wird jedes Mal ein Signal gesendet wenn das Video sich an einer neuen Position befindet. Das erlaubt es, Funktionen aufzurufen wenn sich das Video an bestimmten Stellen befindet. Dadurch wird die Synchronisation zwischen dem Video und Filmdrehbuch-Panel hergestellt. Befindet sich das

³<http://www.qt.io>

5 Implementierung

Video an einem Zeitpunkt, zu dem auch ein Untertitel existiert der zu einem Dialog im Filmdrehbuch zugeordnet wurde, so springt das Filmdrehbuch-Panel zu diesem Dialog.

Für das Filmdrehbuch-Panel wurde eine *QWebView* verwendet. Das Drehbuch wird also als eine eingebettete Webseite angezeigt. Dies erlaubt es über die sogenannte *QWebBridge* den C++ Teil des Codes mit JavaScript zu verbinden. Erst dadurch wird es möglich zu einer bestimmten Stelle des Drehbuchs zu springen, da JavaScript Funktionen anbietet, zu bestimmten Elementen einer Webseite zu scrollen. Auch erlaubt die Benutzung der *QWebView* interaktive Elemente in das Filmdrehbuch-Panel zu platzieren. So war es möglich Buttons neben die Dialoge zu setzen, die bei Auswahl zur jeweiligen Stelle im Video springen. Ein weiterer Grund für den Einsatz der *QWebView* ist die Benutzung von CSS für das Styling des Drehbuchs, wie zum Beispiel für die farbliche Hervorhebung der Filmcharaktere.

Da die im Rahmen des Systems entwickelte Überblicksvisualisierung ein hohes Maß an Anpassbarkeit benötigt, ist es notwendig die einzelnen Elemente der Visualisierung auf einem niedrigen Abstraktionslevel zu definieren. Hierfür eignet sich in Qt die *QGraphicsScene* zusammen mit der *QGraphicsView*. Die *QGraphicsScene* bietet eine Oberfläche für die Darstellung von zweidimensionalen graphischen Elementen, den sogenannten *QGraphicsItems*. Dazu zählen beispielsweise Rechtecke, Ellipsen, Text oder Bilder. Es ist außerdem möglich eigene Elemente zu definieren. Um eine Interaktion mit Elementen der Visualisierung herzustellen, müssen die Basisklassen für Rechtecke oder der anderen Elemente erweitert werden, und Funktionen für die Mausinteraktionen überladen werden.

Die *QGraphicsItems* können in einer Eltern-Kind Beziehung zu anderen *QGraphicsItem* stehen. Dadurch werden ihre Koordinatensysteme verknüpft und es entsteht ein Szenengraph der einzelnen graphischen Elemente. Wird also ein Elternobjekt verschoben, so werden alle Kinderobjekte dieses Objekt ebenfalls verschoben. Muss nun eine *SceneView* verschoben werden, da der Benutzer eine weitere Szene angewählt hat, so werden nun die Kinderobjekte der *SceneView* ebenfalls verschoben, und es muss nur die neue Position der zu verschiebenden *SceneView* berechnet werden. Das zeigt die Nützlichkeit der Eltern-Kind Beziehung von *QGraphicsItems* in Qt.

Im System werden die *SceneViews* als eigene *QGraphicsItems* definiert. Ihnen untergestellt sind wiederum Kinderobjekte, für die einzelnen Taglines oder auch die Textlabels für die Namen der Taglines. Die vertikalen Linien und Blöcke der Visualisierung werden mit Hilfe von *QGraphicsRectItems* gezeichnet. Ein *QGraphicsRectItem* stellt ein Rechteck innerhalb einer *QGraphicsScene* dar. Es ist möglich diesem unter anderem eine Position, Höhe, Breite, Farbe und einen Rahmen zugeben. Das ermöglicht es eine detaillierte und anpassbare Visualisierung zu erstellen. Die *QGraphicsView* ermöglicht es eine *QGraphicsScene* in die Anwendung selbst zu integrieren und zu rendern. Außerdem ist die *QGraphicsView* dafür verantwortlich, dass Teile der *QGraphicsScene*, die nicht im ViewPort sichtbar sind, durch eine Scrollbar erreichbar sind.

Insgesamt hat sich Qt als eine nützliches Framework für die Entwicklung der Überblicksvisualisierung herausgestellt. Besonders die leichte Verarbeitung von Videos und die hohe Anpassbarkeit bei der Darstellung von graphischen Elementen in der *QGraphicsScene* sind hier als wichtige Argumente dafür zu nennen.

6 Anwendungsfälle

Im Folgenden werden typische Anwendungsfälle beschrieben, die mit Hilfe der entwickelten Überblicksvisualisierung durchgeführt werden können. Dabei wird zum einen ein tieferer Einblick in die Funktionsweise gegeben und zum anderen die Flexibilität und Nützlichkeit des beschriebenen Ansatzes gezeigt.

6.1 Analyse einer Beziehung von Filmcharakteren

In den meisten Spielfilmen sind die Filmcharaktere die zentralen Elemente. Als Filmcharakter wird dabei jeder benannte Akteur in einem Film betrachtet. Im Rahmen einer Handlung agieren die Filmcharaktere miteinander, indem sie Dialoge führen oder gemeinsam eine Aktivität durchführen. Für die Filmanalyse ist deshalb diese Interaktion von Filmcharakteren ein interessanter Aspekt.

Hier spielen vor allem folgende Fragestellungen eine wichtige Rolle:

- Worüber reden bestimmte Charaktere?
- Wie oft reden bestimmte Charaktere miteinander?
- Wann reden bestimmte Charaktere miteinander?
- In welchen Szenen kommen bestimmte Charaktere gemeinsam vor?

Dazu kommen Fragen auf einer semantischen Ebene, für die es keine exakten Antworten gibt:

- Wie ist die Beziehung bestimmter Charaktere charakterisiert?
- Wie entwickelt sich die Beziehung von Charakteren?
- Welche Motive oder Sinnbilder spielen eine Rolle in der Beziehung von Charakteren?

Diese Fragen sind ohne computergestützte Verfahren nur mühsam zu beantworten. Die im Rahmen dieser Arbeit entwickelte Visualisierung kann jedoch eine effektiven Beantwortung dieser Fragen unterstützen.

Wie genau eine solche Analyse einer Beziehung von Filmcharakteren mit Hilfe des Systems durchgeführt werden kann wird im Folgenden näher erläutert. Das System führt bei der anfänglichen Analyse der Untertitel und des Drehbuchs für jeden Filmcharakter eine Tagline ein. Die Tagline hat den Namen des Filmcharakters und repräsentiert diesen in der gesamten Visualisierung. Es wird außerdem eine Liste aller Taglines angezeigt.

6 Anwendungsfälle

Inception handelt von einer Gruppe von Agenten angeführt von Dom Cobb, die in die Träume von Menschen eindringen können um ihnen Gedanken zu stehlen. Ein japanischer Geschäftsmann beauftragt Cobb eine *Inception*, die Einpflanzung einer Idee, durchzuführen, und bietet Cobb dafür an, dass dieser in sein Heimatland zu seinen Kindern zurückkehren kann. Während der Mission dringen die Agenten in immer tiefere Traumebenen, dabei wird Cobb immer wieder von seiner verstorbenen Frau Mal besucht.

Wir wollen nun die Beziehung von Cobb und Mal mit Hilfe der Überblicksvisualisierung näher untersuchen. Dazu kombinieren wir die Taglines der beiden Charaktere. Zuerst öffnen wir den Dialog zur Kombination von Taglines, woraufhin ein neues Fenster erscheint, auf dem alle Taglines aufgelistet sind (Abbildung 6.1).



Abbildung 6.1: Der Dialog für die Kombination von Taglines. Wir wählen “Cobb“ und “Mal“ aus und bestätigen unsere Auswahl.

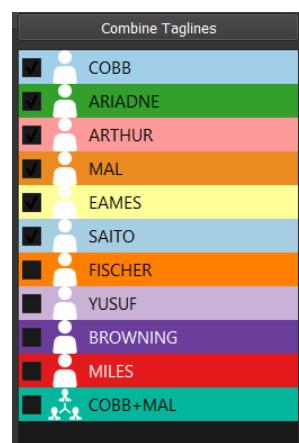


Abbildung 6.2: Durch die Kombination der Taglines “Cobb“ und “Mal“ wurde eine neue Tagline “Cobb + Mal“ generiert.

6.1 Analyse einer Beziehung von Filmcharakteren

Hier wählen wir "Cobb" und "Mal" aus und bestätigen unsere Auswahl (Abbildung 6.1). Nun gelangen wir wieder auf die Hauptansicht. Es wurde eine neue Tagline namens "Cobb + Mal" generiert und zur Liste der Taglines hinzugefügt (Abbildung 6.2). Diese wählen wir aus, woraufhin sie zur Liste der aktiven Taglines hinzugefügt wird und in der MovieView angezeigt wird. In der Tagline-Liste wählen wir nun alle anderen Taglines ab, da wir uns speziell für die Beziehung von Cobb und Mal interessieren. Durch die aggregierte Tagline in der MovieView erhalten wir einen ersten Überblick über die zeitliche Entwicklung der Beziehung zwischen Cobb und Mal.



Abbildung 6.3: Die MovieView für die Charaktere Cobb und Mal aus *Inception*.

Schauen wir uns den Verlauf der Tagline nun genauer in der MovieView (Abbildung 6.3) an: Wie fahren mit der Maus über den Namen der Tagline, woraufhin die einzelnen Balken der Tagline hervorgehoben werden. Durch die Höhe der Balken sehen wir, an welchen Teilen des Films Cobb und Mal auftreten und in welchen Szenen sie genau auftreten. Zu Beginn des Filmes sehen wir zwei Szenen (siehe Bereich A in Abbildung 6.3) in denen die beiden Charaktere mit einer relativ hohen Frequenz auftauchen. Hier handelt es sich möglicherweise um das erste Treffen der beiden Charaktere. Darauf folgen im Verlauf des Films einzelne Szenen in denen die beiden vorkommen. In der Mitte des Films befinden sich zwei Bereiche (B und C in Abbildung 6.3) in denen Cobb und Mal. Besonders interessant ist der Verlauf am Ende der Tagline (Bereich D in Abbildung 6.3). Hier folgen gleich mehrere Szenen mit steigender Häufigkeit aufeinander. Möglicherweise liegt hier eine Art Klimax in der Beziehung der beiden Charaktere.

Eine weitere Besonderheit der Beziehung lässt sich durch die angezeigten Taglines erkennen: Die Tagline von Mal und die kombinierte Tagline von Mal und Cobb haben einen ähnlichen Verlauf. In den Szenen in denen Mal auftritt, tritt in nahezu allen Fällen auch Cobb auf. Andersherum kommt Cobb auch in vielen Szenen vor in denen Mal nicht vorkommt. Das deutet darauf hin, dass Mal eine starke Verbindung zu Cobb hat und ihn "verfolgt". Wir merken uns die interessanten Stellen und wollen nun die Beziehung genauer untersuchen.

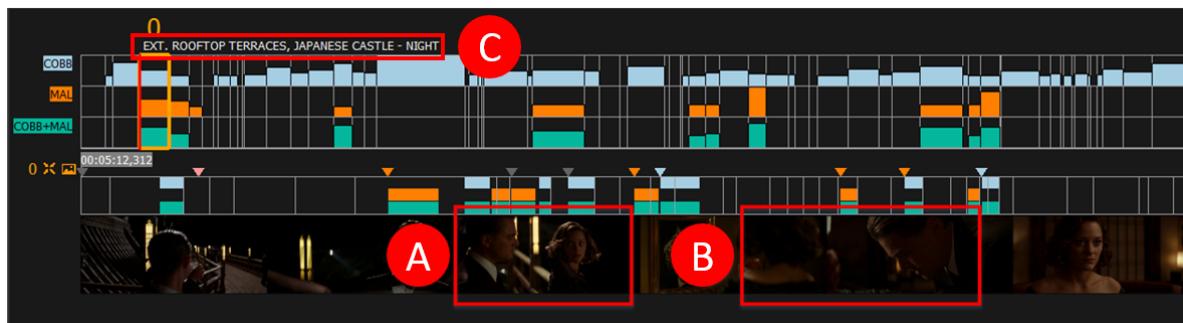


Abbildung 6.4: Die SceneView der Szene des ersten gemeinsamen Auftretens von Cobb und Mal. A und B zeigen Frames des Storyboard, in dem beide Charaktere zu sehen sind.

6 Anwendungsfälle

Wir fahren nun mit der Maus über die erste Szene in der Cobb und Mal gemeinsam auftauchen. Ein Tooltip über der Movieview (C in Abbildung 6.4) verrät uns, dass die Szene sich nachts in einem alten japanischen Schloss abspielt. Wir überfliegen nun das Storyboard innerhalb der SceneView. Schauen wir uns das Storyboard näher an, so können wir Cobb und Mal in zwei Frames (A und B in Abbildung 6.4) erkennen. Offensichtlich findet hier eine Konversation der beiden Charaktere statt.

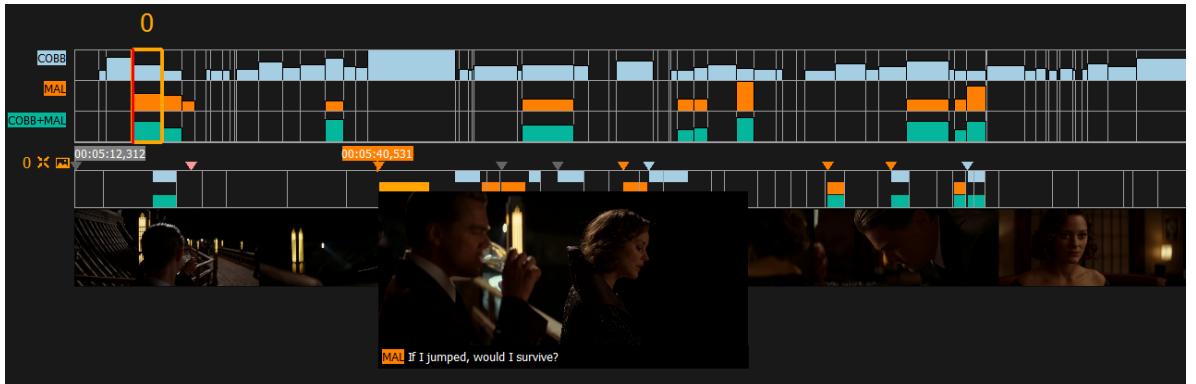


Abbildung 6.5: Um die Konversation näher zu untersuchen, fahren wir mit der Maus über einen Tag. Ein Vorschaubild mit dem zugehörigen Dialog erscheint daraufhin.

Wir wollen diese Konversation nun untersuchen. Aus der MovieView wissen wir, dass Mal die Farbe Orange zugewiesen wurde. Deshalb fahren wir über den ersten orangenen Tag der SceneView, woraufhin ein Vorschaubild mit dem Dialog an dieser Stelle erscheint (Abbildung 6.5).

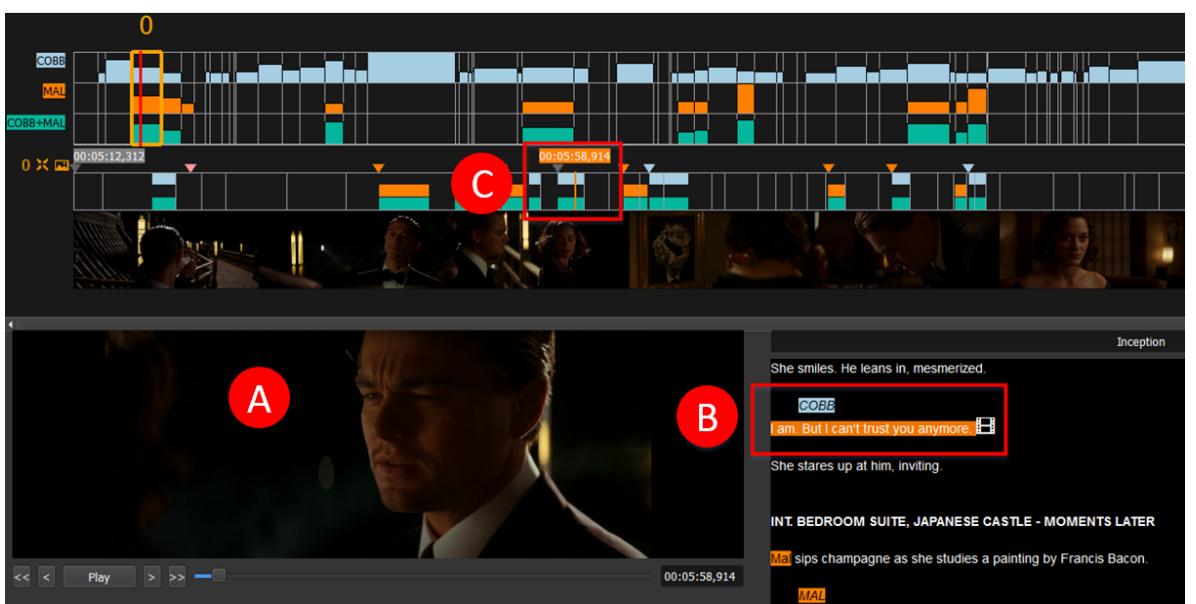


Abbildung 6.6: Während der Videoplayer das Video abspielt, bleiben die Visualisierung und das Filmdrehbuch synchron zu einander.

6.2 Exploration von Filmstücken zu einem Suchbegriff

Wir klicken nun auf den Tag und starten den Videoplayer um diese Konversation anzuschauen. Wir sehen das eigentliche Video unterhalb der SceneView (A in Abbildung 6.6). Gleichzeitig wird die entsprechende Textstelle im Filmdrehbuch-Panel markiert (B in Abbildung 6.6), die wir für unsere Analyse durchlesen können. Die Visualisierung zeigt uns über einen Zeitstempel (C in Abbildung 6.6) stets unsere aktuelle zeitliche Position in der Szene an.

Wir erfahren aus der Konversation, dass ein Konflikt zwischen Cobb und Mal existiert, und dass Cobb Mal nicht mehr Vertrauen kann. Wir wiederholen dieses Vorgehen nun für die weiteren Szenen, in denen wir besonders interessante Szenen bezüglich der Charaktere auswählen und näher untersuchen. Dadurch erfahren wir, dass Mal bereits gestorben ist und lediglich eine Illusion im Unterbewusstsein von Cobb ist, da dieser sich für den Tod von Mal verantwortlich fühlt.

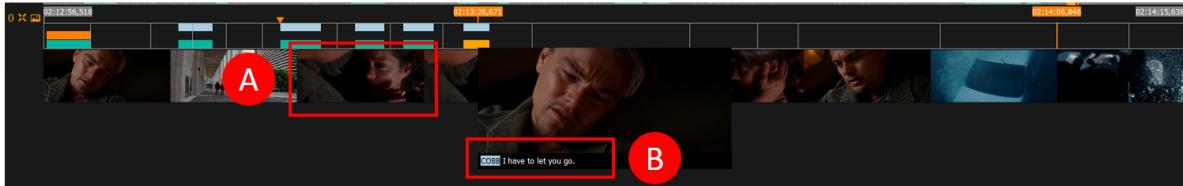


Abbildung 6.7: Eine Schlüsselszene in der Beziehung von Cobb und Mal.

Zuletzt wählen wir die letzte Szene (Abbildung 6.7) aus, in der beide Hauptcharaktere vorkommen. Solche Szenen am Ende eines Film haben neben dem ersten Vorkommen oft eine besondere Bedeutung, da hier möglicherweise der Höhepunkt eines Films liegt. Eine Betrachtung des Storyboards zeigt, dass wir eine relevante Szene entdeckt haben. Wir sehen Mal, wie sie weinend in den Armen von Cobb liegt (A in Abbildung 6.7). Eine Interaktion mit den Tags zeigt uns schnell den Inhalt ihrer letzten Konversation (B in Abbildung 6.7). Cobb akzeptiert endlich den Tod seiner Frau und wird von seinen Schuldgefühlen befreit. Wie haben so eine der Schlüsselszenen der Beziehung der beiden Charaktere schnell finden und analysieren können.

Durch diesen Anwendungsfall zeigen sich bereits die Nutzen der Visualisierung:

- Wir können schnell einen Überblick über den gesamten Verlauf der Beziehung gewinnen.
- Filmstellen mit einer potentiellen Relevanz für die Analyse können effizient gefunden werden.
- Bei Interesse können wir spezielle Filmstellen näher untersuchen.

6.2 Exploration von Filmstücken zu einem Suchbegriff

Ein wichtiger Anwendungsfall für die Filmanalyse ist die Suche nach bestimmten Ereignissen in denen ein bestimmter Suchbegriff erwähnt wird. Angenommen wir analysieren die Herr der Ringe Trilogie nach der Bedeutung und Relevanz des titelgebenden Ringes. Um eine nähere Analyse dieser Art durchzuführen, müssen zuerst die relevanten Szenen gefunden werden. Ohne ein Tool das auf eine solche Analyse ausgerichtet ist, muss das gesamte Video nach Stellen gesucht werden, in denen der Ring eine Rolle spielt oder erwähnt wird.

6 Anwendungsfälle

Das vorgestellte System kann bei der Suche nach relevanten Filmstellen zu einem bestimmten Suchbegriff helfen. Zu diesem Zweck können Suchanfragen an das System gestellt werden. Dabei werden die Texte aller bestehenden Taglines nach dem Suchbegriff durchschaut. Bei Untertiteln wird dabei der Text der Untertitel für die Suche benutzt. Dadurch werden alle Filmstellen in denen der Suchbegriff erwähnt wird aufgefunden. Bei benutzerdefinierten Taglines werden die Notizen für die Suche herangezogen. Wurde die Suche erfolgreich durchgeführt so wird für das Suchergebnis eine eigene Tagline generiert. Diese enthält alle Tags die den Suchbegriff enthalten.

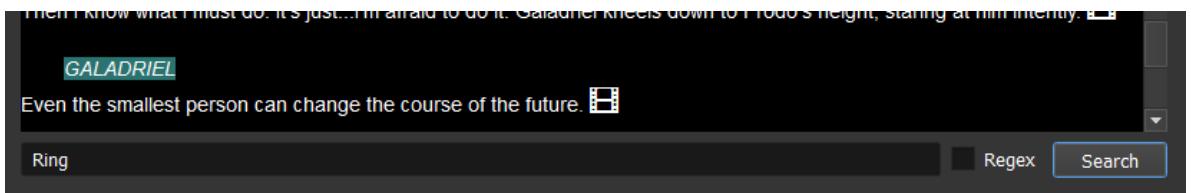


Abbildung 6.8: Der Benutzer sucht nach dem Begriff “Ring“.

Angenommen wir wollen den ersten Teil der Herr der Ringe Trilogie nach dem Begriff “Ring“ durchsuchen. Dafür gibt der Benutzer zuerst den Suchbegriff wie in Abbildung 6.8 in das Suchfenster unterhalb des Filmdrehbuch-Panels ein. Daraufhin beginnen wir die Suche durch einen Klick auf *Suchen*.

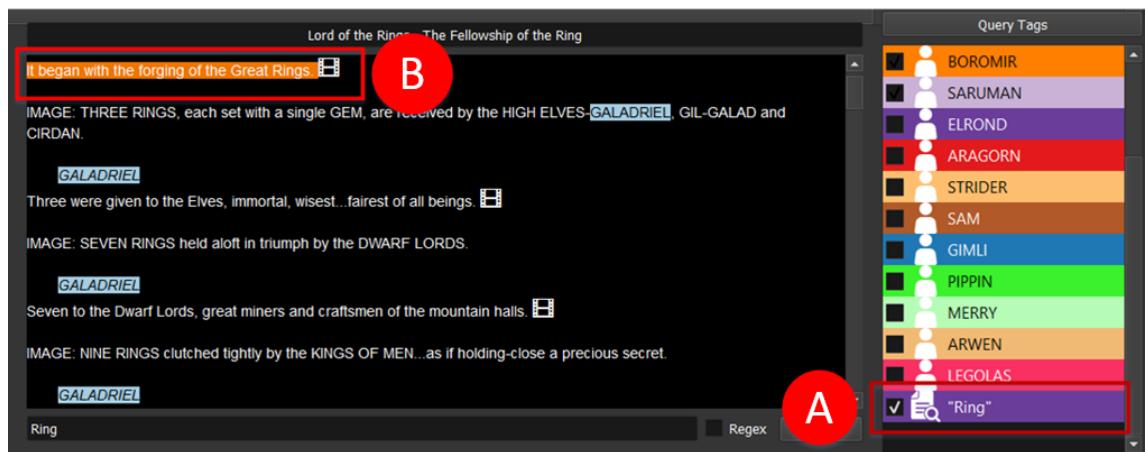


Abbildung 6.9: Eine neue Tagline namens “Ring“ erscheint in der Liste (A) und eine Textstelle die den Suchbegriff enthält wird hervorgehoben (B).

Es erscheint nun, wie in Abbildung 6.9 zu sehen, eine neue Tagline in der Liste der Taglines, neben dem Filmdrehbuch-Panel. Diese Tagline erhält den Namen des Suchbegriffs, in diesem Fall “Ring“. Das Texticon neben dem Namen signalisiert, dass es sich hierbei um eine Tagline eines Suchbegriffs handelt. Die Tagline wird außerdem automatisch ausgewählt, wodurch sie zu den aktiven Taglines hinzugefügt und in der MovieView angezeigt wird. Gleichzeitig zeigt das Filmdrehbuch-Panel das erste Vorkommen des Suchbegriffs an und hebt die entsprechende Stelle farblich hervor (B in Abbildung 6.9).

6.2 Exploration von Filmstücken zu einem Suchbegriff



Abbildung 6.10: Die resultierende Tagline für den Suchbegriff "Ring" in der MovieView. Die markierten Bereiche A - D stellen besonders relevante Filmabschnitte in Bezug auf den Ring dar.

Der Benutzer kann nun mit der eigentlichen Suche nach relevanten Filmstücken zu dem Suchbegriff "Ring" beginnen. Dazu betrachten wir nun die in Abbildung 6.10 gezeigte Tagline "Ring" in der MovieView. So erhält der Benutzer schnell einen Überblick in welchen Szenen des Films der Ring eine Rolle spielt und wie stark die Wichtigkeit in dieser Szene ist. Es ist erkennbar, dass der Ring vor allem in einer langen Szene zu Beginn (Bereich A), in zwei längeren Bereichen in der Mitte des Filmes (Bereich B und C), und wieder am Ende des Films (Bereich D) eine wichtige Rolle spielt.

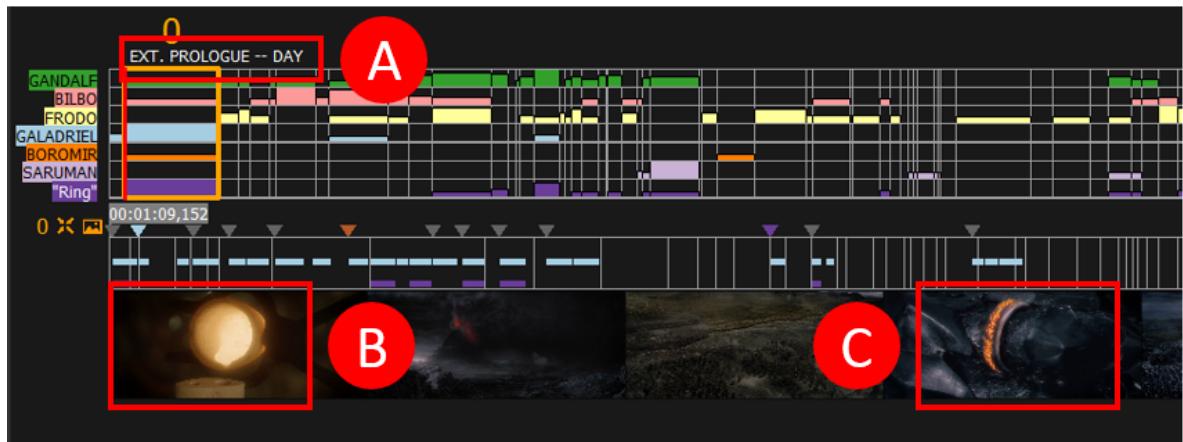


Abbildung 6.11: Ein Überblick des Prologs. A: Die Szenenbeschreibung. B und C: Relevante Frames bezüglich des Rings.

Szenen in der Tagline bei denen die Balkenhöhe der Tags besonders groß ist, deuten auf eine überdurchschnittlich hohe Erwähnung des Rings in diesen Szenen hin. Es bietet sich deshalb an diese Szenen besonders genau zu betrachten. Wählen wir also deshalb die Szene zu Beginn des Films (Bereich A in Abbildung 6.10) aus. Die SceneView erscheint nun unterhalb der MovieView und wir erhalten einen Detailblick auf diese Szene (siehe Abbildung 6.11). Ein Text über dem ausgewählten Szene in der MovieView verrät, dass es sich hier um den Prolog (A) handelt.

Ausgewählte Frames geben einen ersten Überblick über die Szene. In diesem Fall lassen die Frames darauf schließen, dass in dieser Szene die Erschaffung des Rings beschrieben wird. So zeigt der erste Frame (B) den Vorgang des Gießens, während der vierte Frame (C) einen noch glühenden Ring zeigt. So konnte bereits ein schneller Überblick über die Szene geben werden ohne einen Teil des Films betrachten zu müssen.

Nun kann diese Szene näher untersucht werden. Wie genau und von wem wird der Ring erwähnt?

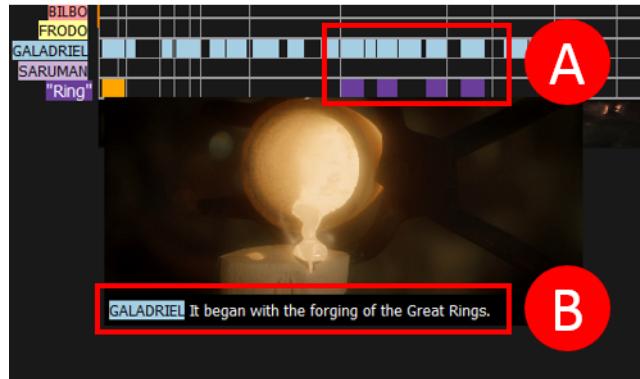


Abbildung 6.12: Ein Auschnitt der SceneView des Prologs.

Um diese Frage zu beantworten, maximieren wir zuerst die SceneView durch einen Mausklick auf die Pfeile am linken Rand der SceneView. Dadurch erscheinen links die Namen der aktiven Taglines. Außerdem wird die Ansicht der Taglines vergrößert und Trennlinien werden zwischen den Taglines erzeugt. Dies führt zu einer verbesserten Übersichtlichkeit und einer angenehmeren Interaktion. Dieser Sachverhalt wird in Abbildung 6.12 dargestellt. Wir sehen, dass der Filmcharakter Galadriel eine besonders wichtige Rolle im Prolog spielt, erkennbar an der hohen Anzahl an Tags in der SceneView, die gleichzeitig mit den Erwähnungen des Rings stattfinden (A in Abbildung 6.12).



Abbildung 6.13: Die Frames mit zugehörigem Untertitel aus der SceneView, in der der Ring erwähnt wird. Hier wird die Geschichte des Rings aus *Herr der Ringe* beschrieben.

Fahren wir nun mit der Maus über einzelne Tags der Ring-Tagline, so erscheint ein Bild der zu dem Tag gehörenden Filmstelle. Darunter wird der Text des Tags angezeigt. In diesem Fall sehen wir die Entstehungsgeschichte des Rings (B in Abbildung 6.12) und wie der Ring von Person zu Person gewechselt ist, bis er schließlich bei Bilbo (Abbildung 6.13) landet.

Dieser Anwendungsfall hat aufgezeigt, wie die Überblicksvisualisierung dazu benutzt werden kann, um schnell einen Überblick über relevante Szenen und Filmstellen für einen bestimmten Suchbegriff zu erhalten.

6.3 Manuelle Annotation

Will ein Benutzer eine bestimmte Fragestellung beantworten, so kann es sein dass die Analyse der Taglines einzelner Filmcharaktere oder eine Textsuche nicht ausreicht. Das ist vor allem dann der Fall, wenn die Fragestellungen nicht direkt mit den Filmcharakteren zusammenhängt oder im Film nur

bildlich dargestellt wird. In einem Film des Genres Western könnten zum Beispiel die Schießereien von Interesse sein. Wann finden Schießereien statt, und welche Charaktere sind daran beteiligt? Die Analyse eines solchen spezifischen, benutzerdefinierten Themas erfordert die Möglichkeit eigene Taglines zu generieren.

Aus diesem Grund ist es möglich benutzerdefinierte Taglines anzulegen. Dafür müssen die Filmstellen, für die sich der Benutzer interessiert, manuell annotiert werden. Die einzelnen Tags der Tagline können durch eigene Notizen ergänzt werden um bei der Analyse einen besseren Überblick zu erhalten. Dieser Vorgang der manuellen Annotation ist mit dem restlichen System und der Überblicksvisualisierung integriert. Durch die Visualisierung und Interaktionsmöglichkeiten wird die Annotation erleichtert.

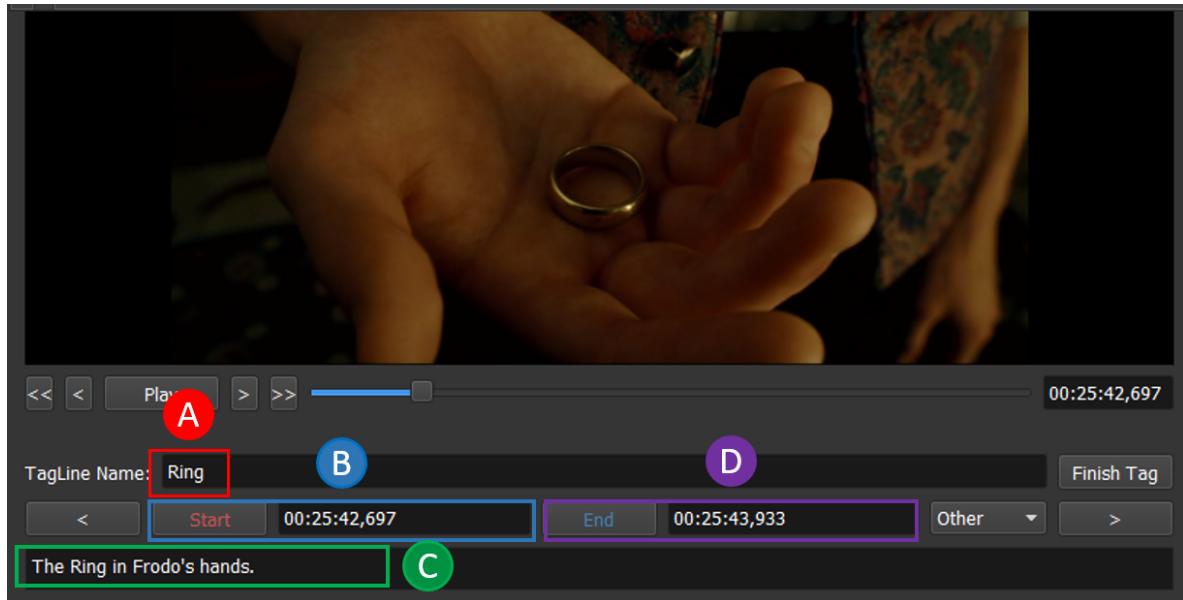


Abbildung 6.14: Die Benutzeroberfläche für die manuelle Annotation. A: Der Name der zu erstellenden Tagline. B: Die Startzeit des aktuellen Tags. C: Optionale Notizen zu dem aktuellen Tag. D: Die Endzeit des aktuellen Tags.

Wir betrachten nun einen konkreten Anwendungsfall, der diese Aspekte näher erläutert. Angenommen wir interessieren uns für alle Filmstellen, an denen der Ring aus Herr der Ringe gezeigt wird. Für die manuelle Annotation gibt es im System die TaggingView (siehe Abbildung 6.14), die es ermöglicht einfach Filmstellen zu annotieren und mit Notizen zu versehen. Zuerst geben wir unserer Tagline einen Namen und geben dieses in das entsprechende Feld (A in Abbildung 6.14) ein. Wir können uns nun beliebig durch das Video bewegen und nach passenden Filmstellen suchen. So können wir die Tagline der Textsuche aus dem vorhergehenden Abschnitt benutzen um mögliche Filmstellen, in denen der Ring auftaucht, schneller zu finden. Wie klicken also auf die erste Szene, in der der Ring erwähnt wird und betrachten die Szene dann im Videoplayer. Zu dem Zeitpunkt 00:25:42,547 sehen wir nun den Ring. Wir halten das Video über die Kontrolleiste des Videoplayers an und bewegen uns mit Hilfe der Pfeilknöpfe zum ersten Frame zurück, in dem der Ring zu sehen ist. Nun wollen wir unseren ersten Tag erstellen. Wir drücken erst auf den "Start"-Knopf der TagView (B in Abbildung

6 Anwendungsfälle

6.14). Daraufhin wird der Startzeitpunkt des Tags auf die aktuelle Zeit des Videoplayers gesetzt und in einem Texfeld neben dem Startknopf angezeigt.

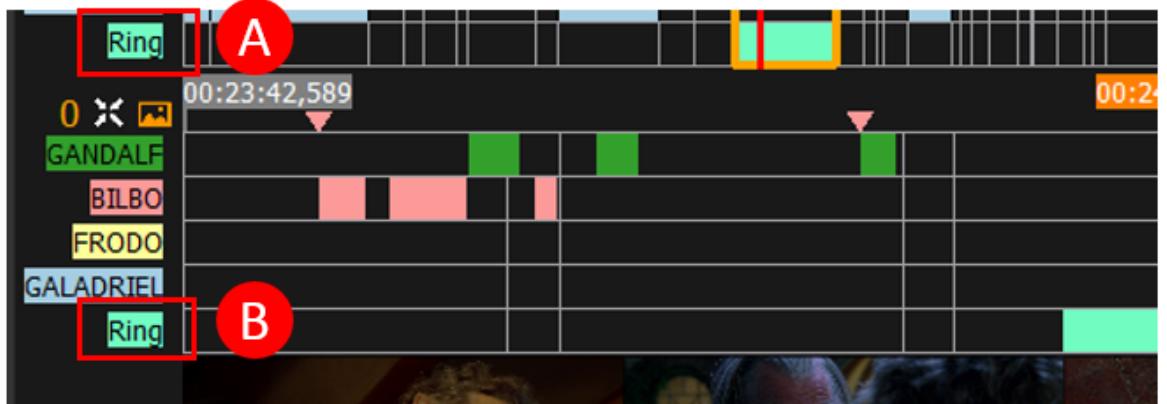


Abbildung 6.15: Eine neue Tagline wird angelegt und in der MovieView (A), sowie in der aktuellen SceneView (B) angezeigt.



Abbildung 6.16: In der maximierten Ansicht der SceneView werden die verfassten Notizen angezeigt.

Wir lassen das Video weiterspielen und halten es an, sobald der Ring nicht mehr zu sehen ist. Nun können wir für den Tag optional eine Notiz aufschreiben. Wir notieren uns "The Ring in Frodo's hands." und erstellen den Tag endgültig, indem wir auf den "Ende"-Knopf (D in Abbildung 6.14) drücken. Dadurch wird die Endzeit des Tags gesetzt. Da wir nun den ersten Tag unserer Tagline erschaffen haben, wird nun die Tagline angelegt und in der MovieView hinzugefügt (Abbildung 6.15). Auch in der SceneView erscheint unser neuer Tag mit der Notiz (Abbildung 6.16).

Dieser Anwendungsfall hat aufgezeigt, wie die manuelle Annotation mit dem entwickelten System durchgeführt werden kann und wie diese in das restliche System integriert ist.

7 Zusammenfassung und Ausblick

In dieser Arbeit wurde eine neuartige, interaktive Überblicksvisualisierung, basierend auf der Verknüpfung von Filmdrehbüchern und Untertiteln eines Films, konzipiert und prototypisch entwickelt.

Zuerst wurde basierend auf den Anforderungen an die Überblicksvisualisierung ein Konzept entwickelt, dass eine effektive und zielgerichtete Filmanalyse ermöglicht. Der Inhalt eines Films wurde durch das Konzept von Taglines abstrahiert und aggregiert. Durch eine Synchronisation von Filmdrehbüchern und Untertiteln wurde eine zeitliche Segmentation durchgeführt und die Dialoge zu konkreten Zeitpunkten zugeordnet. Durch den hierarchischen Aufbau der Visualisierung kann ein schneller Überblick gegeben werden, der eine Analyse auf Szenen-, Shot- und Bildebene ermöglicht. Dabei wurde ein besonderer Fokus auf eine effektive Interaktion und die Verknüpfung von Text- und Bildinhalten gelegt. Basierend auf dem Konzept, wurde ein Prototyp entwickelt, der die geforderten Anforderungen erfüllt und das Konzept somit validiert.

Die entwickelte Überblicksvisualisierung gibt einen guten Überblick über den gesamten Inhalt eines Films. Die Vorkommen von Filmcharakteren und ihre Vorkommenshäufigkeiten in bestimmten Szenen sind leicht erkennbar. Einzelne Szenen können durch den "Overview + Detail"-Ansatz gut im Detail exploriert und analysiert werden. Die Storyboards geben außerdem einen schnellen Einblick über den Bildinhalt einer Szene. Mehrere Szenen können gleichzeitig visualisiert und somit verglichen werden. Durch die Interaktion können relevante Textstellen zu Filmstellen angezeigt werden.

Die Synchronisation von Untertiteln und Dialogen aus einem Filmdrehbuch hat sich als eine große Herausforderung herausgestellt. Eine Evaluation von Ähnlichkeitsmaßen hat ergeben, dass mit Hilfe von gängigen Verfahren aus der Textverarbeitung, wie TF-IDF und der Kosinus-Ähnlichkeit, ein Großteil von Untertitel und Dialogen zugeordnet werden kann. Vor allem kurze Untertitel die nur wenige, häufig vorkommende Worte enthalten, haben ein großes Problem dargestellt. Ein weiteres Problem stellten Szenen dar, die wenig oder keinen Dialog haben, wie beispielsweise Actionszenen. Hier stößt der konzipierte Ansatz an seine Grenzen, da keine Dialoge zugeordnet werden können. Trotz dieser Schwierigkeiten hat sich die Verknüpfung von Untertiteln und Filmdrehbuch jedoch als ein überaus nützlicher Aspekt herausgestellt, der für die semantische Analyse eines Films hilfreich ist.

Die Anwendungsfälle haben gezeigt, dass das entwickelte System Analysten dabei unterstützen kann, Beziehungen von Filmcharakteren zu analysieren. Die Suche hat sich als eine gute Möglichkeit herausgestellt, um Schlüsselszenen zu finden und zu explorieren, in denen ein bestimmter Suchbegriff erwähnt wird. Die manuelle Annotation ermöglicht es, Benutzern eine spezialisierte Analyse durchzuführen und dabei die Vorteile der Überblicksvisualisierung zu erhalten. Abschließend kann festgestellt werden, dass der entwickelte Ansatz für eine interaktive Überblicksvisualisierung, basierend auf einer Verknüpfung von Text- und Bildinhalten, eine wertvolle Methode im Bereich der Videovisualisierungen darstellt.

Ausblick

Während sich diese Arbeit bei der Analyse auf die Verknüpfung von Filmdrehbüchern und Untertiteln beschränkt, sind Ansätze denkbar, die auch den Bildinhalt automatisch analysieren. So könnte eine Gesichtserkennung durchgeführt werden, wodurch das Vorkommen von Charakteren besser erkannt werden könnte. Auch eine Analyse der Tonspur in Verknüpfung mit der Visualisierung wäre eine denkbare Erweiterung des Systems.

Ein weiterer Anknüpfungspunkt ist die Analyse des Filmdrehbuchs. Diese geschieht im Rahmen dieser Arbeit auf rein lexikalischer Ebene. Hier wäre der Einsatz von NLP-Technologien denkbar. Dadurch könnte eine Analyse des Textes auf semantischer Ebene durchgeführt werden. Eine interessante Idee wäre die Sentiment Analysis. Diese ermöglicht eine automatisierte Bewertung eines Textes nach der Stimmung. Die Verknüpfung von Filmdrehbuch und Visualisierung könnte dann dafür benutzt werden, um anzusehen, wie positiv oder negativ die Stimmung im Verlauf des Filmes ist, und wie sich diese entwickelt. Filmstellen mit besonders positiver oder negativer Stimmung könnten mit Hilfe der Überblicksvisualisierung schnell gefunden werden. Auch bei der Synchronisierung von Untertiteln und Dialogen könnte NLP nützlich sein. So könnte eine Ähnlichkeit von Sätzen mit Hilfe von NLP auch auf semantischer Ebene berechnet werden.

Die manuelle Annotation ist im vorgestellten Ansatz sehr einfach konzipiert und erlaubt es nur bestimmte Filmstellen mit einer Start- und Endzeit zu annotieren. Ein Ansatzpunkt wäre, hier auch die Annotation im Filmdrehbuch zu erlauben. So könnten relevante Textstellen im Filmdrehbuch markiert und mit Notizen versehen werden, und anschließend in der Überblicksvisualisierung angezeigt werden. Die Kopplung von Text- und Bildinhalten könnte auch auf diese textuellen Annotationen übertragen werden. Eine weitere Möglichkeit die manuelle Annotation zu erweitern wäre es, den Benutzer einen bestimmten Abschnitt von Frames annotieren zu lassen. Damit könnten Objekte oder Personen innerhalb eines Frames annotiert und gekennzeichnet werden, während im jetzigen System nur ganze Frames annotiert werden können.

Insgesamt wäre ein generischer Ansatz denkbar, der nicht speziell auf Spielfilme mit Filmdrehbuch ausgelegt ist, sondern auch für andere Arten von Filmen wie Dokumentationen oder beliebige Arten von Videos anwendbar ist. Dafür könnten verschiedene Arten von textuellen und semantischen Beschreibungen für die Analyse einbezogen werden und mit Bildverarbeitungsalgorithmen kombiniert werden.

Literaturverzeichnis

- [AHS08] P. Achananuparp, X. Hu, X. Shen. The evaluation of sentence similarity measures. In *Data warehousing and knowledge discovery*, S. 305–316. Springer, 2008. (Zitiert auf den Seiten 19, 22 und 65)
- [AMST11] W. Aigner, S. Miksch, H. Schumann, C. Tominski. *Visualization of time-oriented data*. Springer Science & Business Media, 2011. (Zitiert auf den Seiten 7, 15, 16, 17 und 18)
- [BCD⁺12] R. Borgo, M. Chen, B. Daubney, E. Grundy, G. Heidemann, B. Höferlin, M. Höferlin, H. Leitte, D. Weiskopf, X. Xie. State of the Art Report on Video-Based Graphics and Video Visualization. *Computer Graphics Forum*, 31(8):2450–2477, 2012. URL <http://dx.doi.org/10.1111/j.1467-8659.2012.03158.x>. (Zitiert auf den Seiten 7, 13 und 25)
- [BR96] J. S. Boreczky, L. A. Rowe. Comparison of video shot boundary detection techniques. *Journal of Electronic Imaging*, 5(2):122–128, 1996. URL <http://dx.doi.org/10.1117/12.238675>. (Zitiert auf Seite 14)
- [CLH12] T. Chen, A. Lu, S.-M. Hu. Visual storylines: Semantic visualization of movie sequence. *Computers & Graphics*, 36(4):241–249, 2012. (Zitiert auf den Seiten 7 und 28)
- [CMS99] S. K. Card, J. D. Mackinlay, B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999. (Zitiert auf den Seiten 33 und 35)
- [CRCV08] B. Caldwell, L. G. Reid, M. Cooper, G. Vanderheiden. Web Content Accessibility Guidelines (WCAG) 2.0. W3C recommendation, W3C, 2008. <Http://www.w3.org/TR/2008/REC-WCAG20-20081211/>. (Zitiert auf Seite 54)
- [DAE96] A. Dailianas, R. B. Allen, P. England. Comparison of automatic video segmentation algorithms. In *Photonics East’95*, S. 2–16. International Society for Optics and Photonics, 1996. (Zitiert auf den Seiten 13 und 14)
- [Hua08] A. Huang. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, S. 49–56. 2008. (Zitiert auf Seite 19)
- [JBMC10] H. Jänicke, R. Borgo, J. Mason, M. Chen. SoundRiver: Semantically-Rich Sound Illustration. In *Computer Graphics Forum*, Band 29, S. 357–366. Wiley Online Library, 2010. (Zitiert auf den Seiten 7 und 30)
- [KC01] I. Koprinska, S. Carrato. Temporal video segmentation: A survey. *Signal processing: Image communication*, 16(5):477–500, 2001. (Zitiert auf Seite 13)

- [LFK⁺13] S. Lin, J. Fortuna, C. Kulkarni, M. Stone, J. Heer. Selecting Semantically-Resonant Colors for Data Visualization. *Computer Graphics Forum*, 32(3pt4):401–410, 2013. URL <http://dx.doi.org/10.1111/cgf.12127>. (Zitiert auf Seite 52)
- [Mac86] J. Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *ACM Trans. Graph.*, 5(2):110–141, 1986. URL <http://doi.acm.org/10.1145/22949.22950>. (Zitiert auf Seite 38)
- [Mat] Matroska. SRT Subtitles. <http://www.matroska.org/technical/specs/subtitles/srt.html>. (Zitiert auf Seite 57)
- [Mor04] N. Moraveji. Improving Video Browsing with an Eye-tracking Evaluation of Feature-based Color Bars. In *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL ’04, S. 49–50. ACM, New York, NY, USA, 2004. URL <http://doi.acm.org/10.1145/996350.996363>. (Zitiert auf den Seiten 7 und 31)
- [MT99] H. Müller, E. Tan. Movie maps. In *Information Visualization, 1999. Proceedings. 1999 IEEE International Conference on*, S. 348–353. IEEE, 1999. (Zitiert auf Seite 29)
- [PD01] D. Ponceleon, A. Dieberger. Hierarchical brushing in a collection of video data. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, S. 8–pp. IEEE, 2001. (Zitiert auf den Seiten 7 und 27)
- [Ram03] J. Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*. 2003. (Zitiert auf den Seiten 20 und 23)
- [RS03] Z. Rasheed, M. Shah. Scene detection in Hollywood movies and TV shows. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, Band 2, S. II–343. IEEE, 2003. (Zitiert auf Seite 14)
- [The] The Writers Store. How to Write a Screenplay: Script Example & Screenwriting Tips. URL <https://www.writersstore.com/how-to-write-a-screenplay-a-guide-to-scriptwriting/>. (Zitiert auf Seite 59)
- [YYL96] M. Yeung, B.-L. Yeo, B. Liu. Extracting story units from long programs for video browsing and navigation. In *Multimedia Computing and Systems, 1996., Proceedings of the Third IEEE International Conference on*, S. 296–305. IEEE, 1996. (Zitiert auf Seite 13)

Alle URLs wurden zuletzt am 12. 10. 2015 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift