Seminar SS 2002

Multimediale Informationssysteme

Audio / Video Retrieval

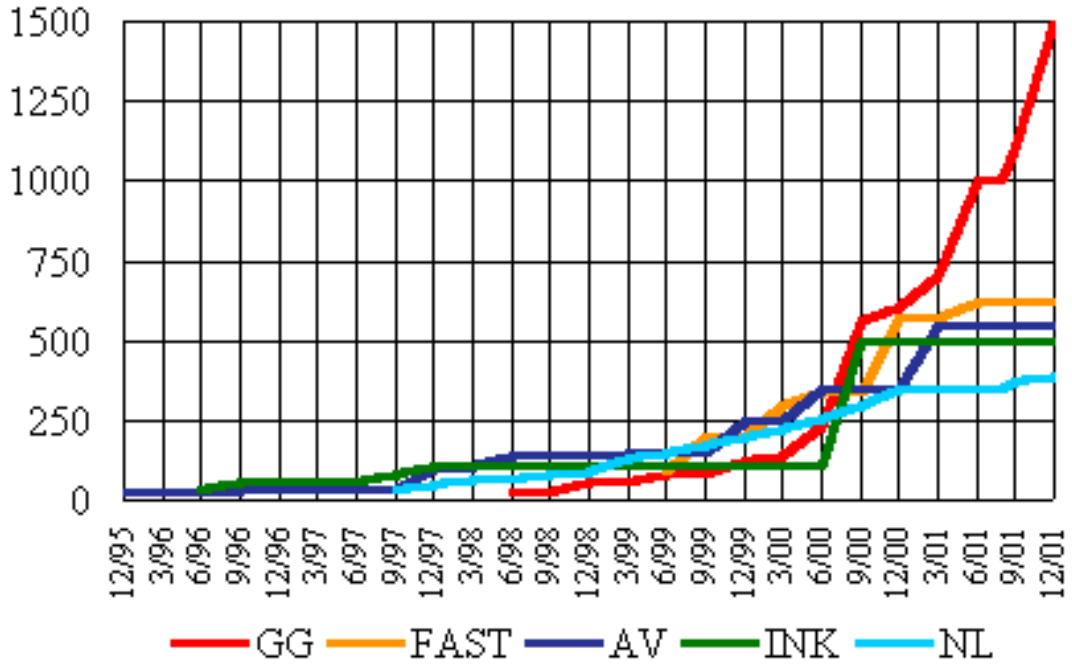
AG Datenbanken und Informationssysteme,
Prof. Dr. Härder, Universität Kaiserslautern

Übersicht

Video-Retrieval
- Beispiel-Video
- Organisation
- Datenbankanfragen
- Frame Segment Trees
- Automatische Indexierung

Audio-Retrieval
- Suche auf Metadaten
- Suche mit akustischen Merkmalen
- Suche mit Noten
- Suche in gesprochenem Text

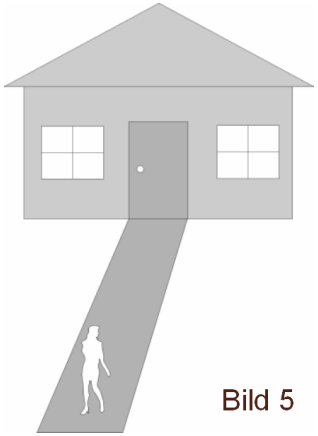
Einleitung



Mehr als drei Milliarden Internetseiten
Täglich kommen ca. sieben Millionen Seiten hinzu

Der Anstieg der Audio- und Videodaten verhält sich gleich

Beispiel-Video



Organisation (1)

Definiton 1:

Ein Paar `(ename,wert)` ist eine *Eigenschaft*. `ename` ist der Name der *Eigenschaft*, `wert` ist eine Menge. Eine Instanz einer *Eigenschaft* lautet `ename = v, v`$\in$ `wert`

Beispiel:

(Nummernschild,X) beschreibt ein Nummernschild. X ist eine Menge, Beinhaltend *Ort x* $Alpha_2$ *x* Num_4

(Größe,R^+*)* beschreibt in positiven reellen Zahlen die Größe einer Person.

Organisation (2)

Definiton 2:

Eine *Objektbeziehung* in ein Paar `(fd,fi)` bei dem gilt:

1. `fd` ist eine Menge von bildabhängigen Eigenschaften
2. `fi` ist eine Menge von bildunabhängigen Eigenschaften
3. `fi≠fd`

Beispiel:

Eine bildabhängige Eigenschaft ist Hemdfarbe.

⟶ Die Farbe ändert sich durch Schatten, Lichteinstrahlung, etc.

Organisation (3)

Definiton 3:

Eine *Objektinstanz* ist ein Tripel `(oid,os,ip).` Es gilt:

1. `oid` ist eine eindeutige Zuweisung für die *Objektinstanz* (Objekt-ID)
2. `os = (fd,fi)` ist eine *Objektbeziehung*
3. `ip` ist eine Menge von Aussagen, bei denen gilt:
   1. Für jede *Eigenschaft* `(ename,wert)` in `fi` enthält `ip` mindestens einen Inhalt von `(ename,wert)`
   2. Für jede *Eigenschaft* `(ename,wert)` in `fd` und jedem Bild *b* des Videos enthält `ip` mindestens einen Inhalt von `(ename,wert)`

Beispiel:

Die Objektinstanz $(\texttt{Person}, \texttt{fd}_1, \texttt{fi}_1)$ hat folgende Inhalte:

`Person` hat als Inhalt *Stefan F.*

`fd` hat als Inhalt (*hat,Koffer)* (Stefan F. hat den Koffer in Bild 1)

`fi` hat als Inhalt (*Größe,187)* (Stefan F. ist 187cm groß)

Organisation – Ergebnis

Bild	Objekt	Bildabhängige Eigenschaften
1	Stefan F. Dope‘s Haus Koffer	besitzt(Koffer), wo(Gehweg_Anfang) Tür(geschlossen)
2	Stefan F. Jens Dope Dope‘s Haus Koffer	besitzt(Koffer), wo(Gehweg_Mitte) wo(in_Tür) Tür(offen)
3	Stefan F. Jens Dope Dope‘s Haus Koffer	besitzt(Koffer), wo(Gehweg_Ende) wo(in_Tür) Tür(offen)
4	Stefan F. Jens Dope Dope‘s Haus Koffer	wo(Gehweg_Mitte) besitzt(Koffer), wo(in_Tür) Tür(offen)
5	Stefan F. Dope‘s Haus Koffer	wo(Gehweg_Anfang) Tür(geschlossen)

Objekt	Bildunabhängige Eigenschaften	Inhalt
Stefan F.	Alter Größe	27 187 cm
Dope‘s Haus	Adresse Art Farbe	Gottlieb-Daimler-Str. 1 67663 Kaiserslautern Holzhaus Braun
Jens Dope	Alter Größe	28 189 cm
Koffer	Farbe Breite Höhe	Rot 40 cm 30 cm

Organisation (4)

Definiton 4:

Eine *Aktivitätsbeziehung* `AktBez` ist eine endliche Menge.
Wenn `(ename,wert`$_1$`)` und `(ename,wert`$_2$`)` in `AktBez` sind,
dann gilt: `wert`$_1$ `= wert`$_2$

Beispiel:

Die Aktivitätsbeziehung *übergeben* hat eine 3-Paar-Beziehung:

1. `(Geber,Person):` Die *Eigenschaft* `Geber` ist vom Typ `Person` (Stefan F.)
2. `(Empfänger,Person):` Die *Eigenschaft* `Empfänger` ist vom Typ `Person` (Jens Dope)
3. `(Gegenstand,Objekt):` Beinhaltet den Wert des `Gegenstands` (Koffer)

Organisation (5)

Definiton 5:

Eine *Aktivität* ist ein Paar, beeinhaltend

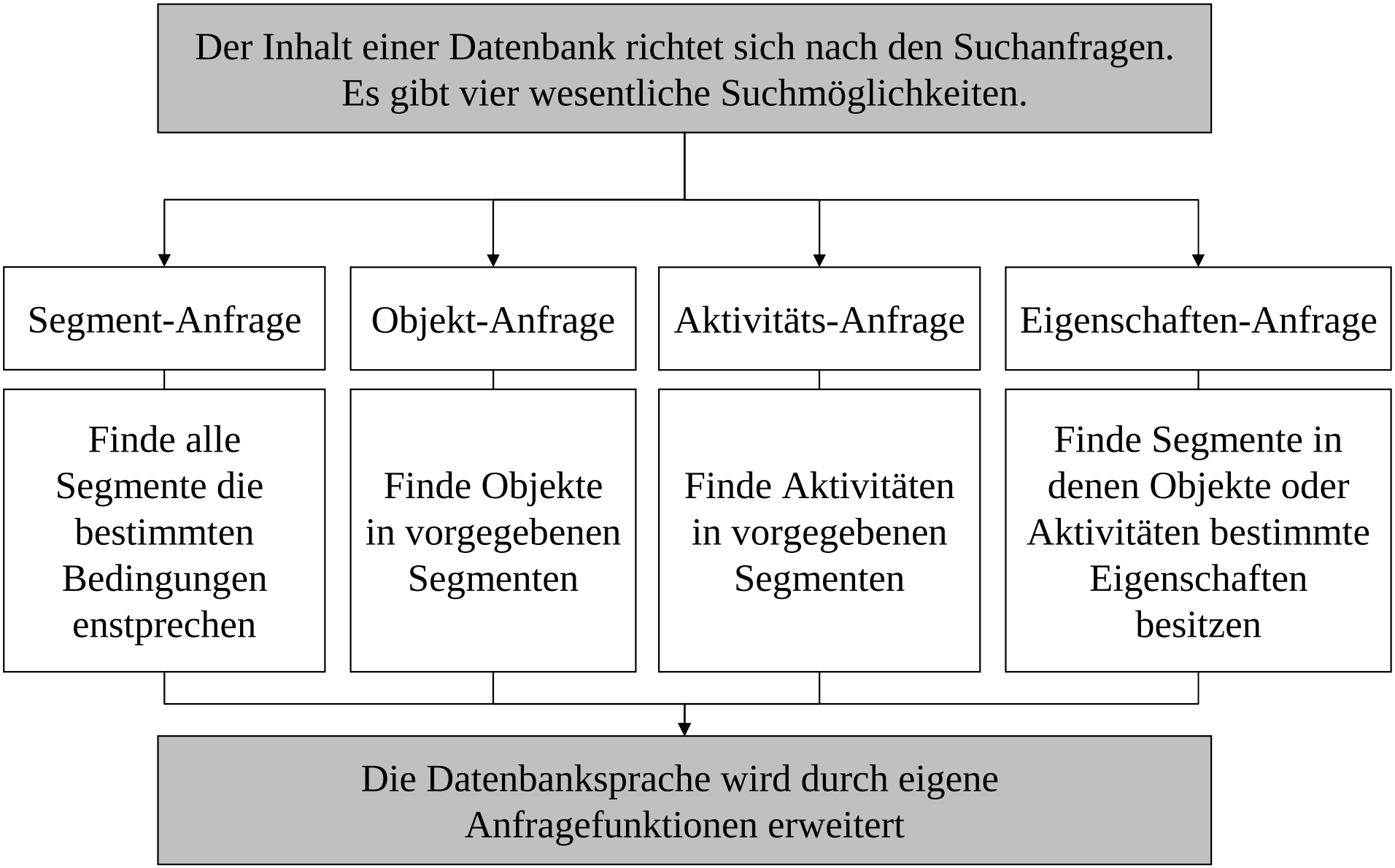
1. `AktID` (der Name der Aktivitätsbeziehung `AktBez`)
2. Für jedes Paar `(ename,wert)` $\in$ `AktBez` gilt: `ename = v, v` $\in$ `wert`

Beispiel *Objektaustausch*:

Die *Aktivität* hat drei Paare `(ename,wert):`

`{(Geber,Person),(Empfänger,Person),(Gegenstand,Objekt)}`

Datenbankanfragen (1)



Datenbankanfragen (2)

1. FindeVideoMitObjekt(o):

Bei der Übergabe eines Objekts o an die Funktion erhält man ein Tripel der Form:
(Video-ID, Start-Bild, End-Bild)

Beispiel: FindeVideoMitObjekt(Koffer)
Ergebnis: (ÜV , 0 , 3000)

Ebenso: 2. FindeVideoMitAktivität(a)

3. FindeObjekteInVideo(v,s,e):

Bei der Übergabe eines Videos v, eines Start-Bildes s und eines End-Bildes e erhält man eine Menge von Objekten.

Beispiel: FindeObjekteInVideo(ÜV,2500,3250)
Ergebnis: {(Stefan F.),(Jens Dope),(Koffer),(Dope‘s Haus)}

Ebenso: 4. FindeAktivitätenInVideo(v,s,e)

Datenbankanfragen (3)

5. FindeVideoMitAktivitätUndEigenschaften(a,e,z):

Bei der Übergabe einer Aktivität a, einer Eigenschaft e und deren Wert z erhält man ein Tripel der Form: (Video-ID , Start-Bild , End-Bild)

Beispiel: FindeVideoMitAktivitätenUndEigenschaften(Sprechen,Person,‘Stefan F.‘)
Ergebnis: (ÜV , 2200 , 2500)

Ebenso: 6. FindeVideoMitObjektUndEigenschaft(a,e,z)

7. FindeAktivitätenUndEigenschaftenInVideo(v,s,e):

Bei der Übergabe eines Videos v und eines Segments [s,e] erhält man eine Menge der Form: Aktivitätsname:Eigenschaft1=Wert1; Eigenschaft2=Wert2; ...; Eigenschaftk=Wertk

Beispiel: FindeAktivitätenUndEigenschaftenInVideo(ÜV,1500,2000)
Ergebnis: übergeben:Empfänger=‚Stefan F.‘;Geber=‚Jens Dope‘;Gegenstand=‚Koffer‘

Ebenso: 8. FindeObjekteUndEigenschaftenInVideo(v,s,e)

Datenbankanfragen – SQL (1)

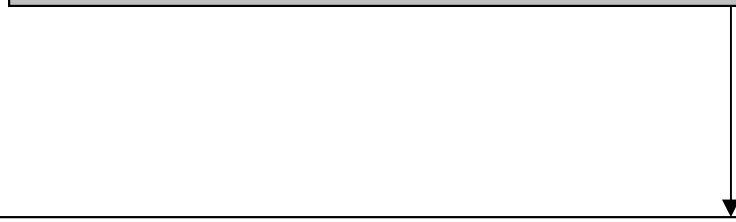
Um die vorgenannten Funktionen zu nutzen muß die Datenbanksprache SQL erweitert werden.

SELECT enthält zusätzlich:	FROM enthält zusätzlich:	WHERE enthält zusätzlich:
Video-ID : [s,e]	video : <quelle><V>	Ausdruck IN Funktionsaufruf

Video-ID: Eindeutige Kennung eines Videos
[s,e] : Segment mit Start-Bild s und End-Bild e
quelle : Videobibliothek
V : Laufvariable über die Videos in der Videobibliothek

Datenbankanfragen – SQL – Beispiele (1)

Finde alle Videos und Segmente aus der Videobibliothek ‚videobib', in denen Jens Dope und Stefan F. zu sehen sind.



```
SELECT vid:[s,e]
FROM   video:videobib
WHERE  (vid,s,e) IN FindeVideoMitObjekt(,Jens Dope') AND
       (vid,s,e) IN FindeVideoMitObjekt(,Stefan F.')
```

Datenbankanfragen – SQL – Beispiele (2)

Finde alle Videos und Segmente aus der Videobibliothek ‚videobib', in denen Jens Dope einen Koffer von Stefan F. erhält.

```
SELECT vid:[s,e]
FROM   video:videobib
WHERE  (vid,s,e) IN FindeVideoMitObjekt('Jens Dope') AND
       (vid,s,e) IN FindeVideoMitObjekt('Stefan F.') AND
       (vid,s,e) IN FindeVideoMitAktivitätUndEigenschaft
                    (Austausch,Objekt,Koffer) AND
       (vid,s,e) IN FindeVideoMitAktivitätUndEigenschaft
                    (Austausch,Empfänger,'Jens Dope') AND
       (vid,s,e) IN FindeVideoMitAktivitätUndEigenschaft
                    (Austausch,Geber,'Stefan F.')
```

Frame Segment Tree (1)

Für jede Aktivität und jedes Objekt eines Bildes erfolgt ein Eintrag in der Datenbank

ineffizient

Speicherung nach Sequenzen

Auftreten interessanter Objekte im Überwachungsvideo					
ID	**Objekt**	**Sequenz**	**ID**	**Objekt**	**Sequenz**
1	Jens Dope	0 – 125	4	Unbekanntes Auto	1000 – 1125
1	Jens Dope	250 – 3000	4	Unbekanntes Auto	1750 – 1875
2	Stefan F.	0 – 4000	4	Unbekanntes Auto	3000 – 3250
3	Koffer	0 – 3000	5	Passant	1500 – 2000
4	Unbekanntes Auto	500 – 625	5	Passant	2750 – 3250

Frame Segment Tree (2)

Aufbau des binären Frame-Segment-Trees:

1. Jeder Knoten im Baum repräsentiert ein Segment des gesamten Videos.

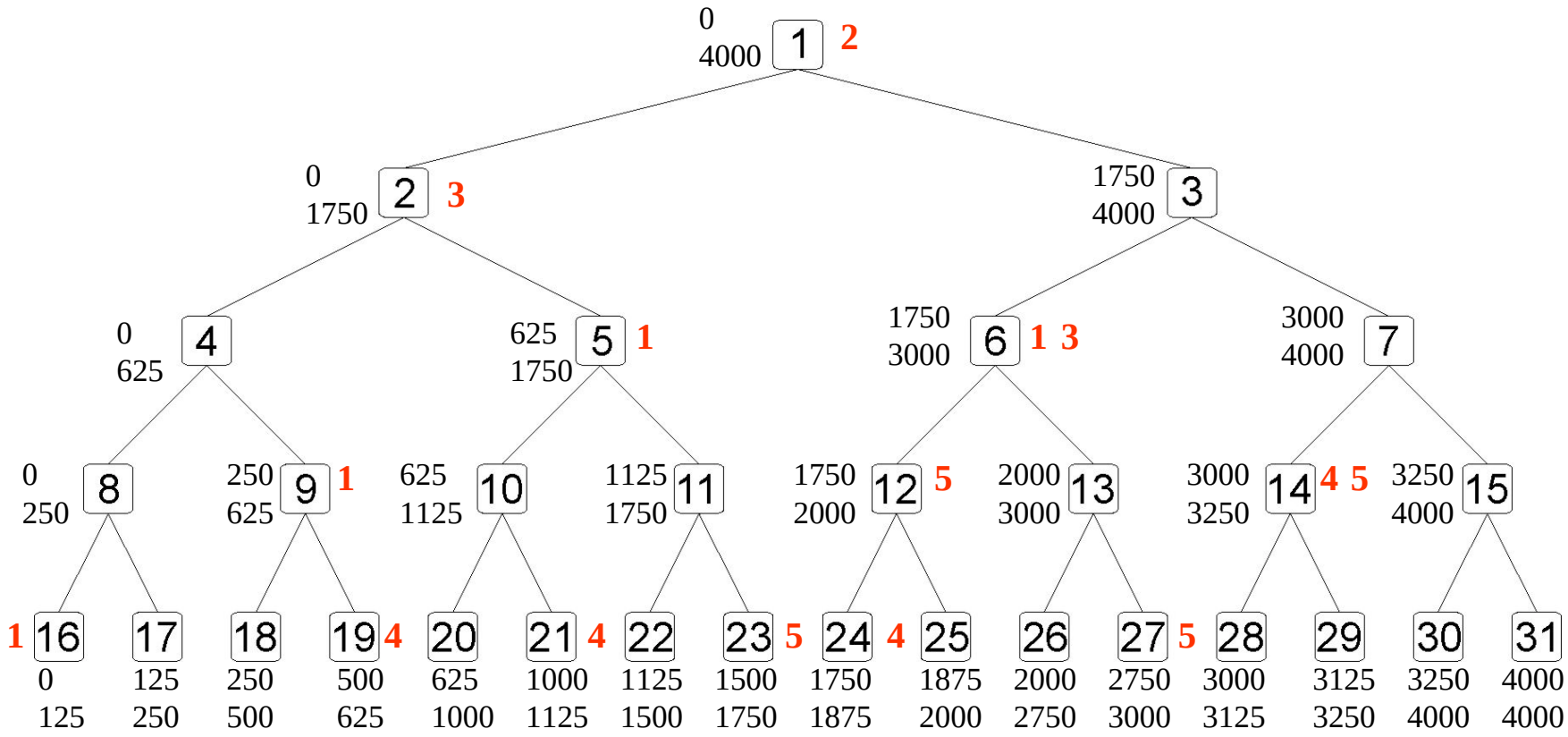
2. Jedes Blatt befindet sich auf der untersten Ebene. Das linke Blatt bezeichnet das Intervall $[s_1,s_2)$, das nächste das Intervall $[s_2,s_3)$, usw.
   Wenn N ein Knoten zwei Kindknoten mit den Intervallen $[p_1,p_2)$ und $[p_2,p_3)$ hat, dann bezeichnet der Knoten N das Intervall $[p_1,p_3)$.

3. Jede Zahl in den Knoten wird als Adresse angesehen.

4. Die Zahl neben den Knoten bezeichnet die ID der Objekte.

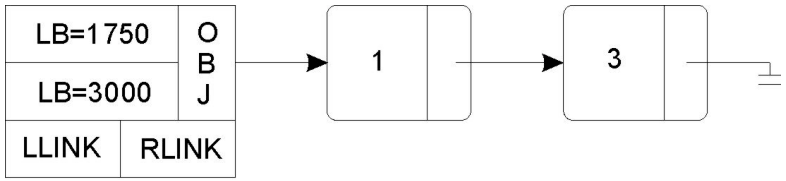
Für die Konstruktion werden alle Anfangs- und Endbilder in einer Liste gespeichert. Diese Liste wird von Duplikaten befreit und sortiert.

Frame Segment Tree (3)

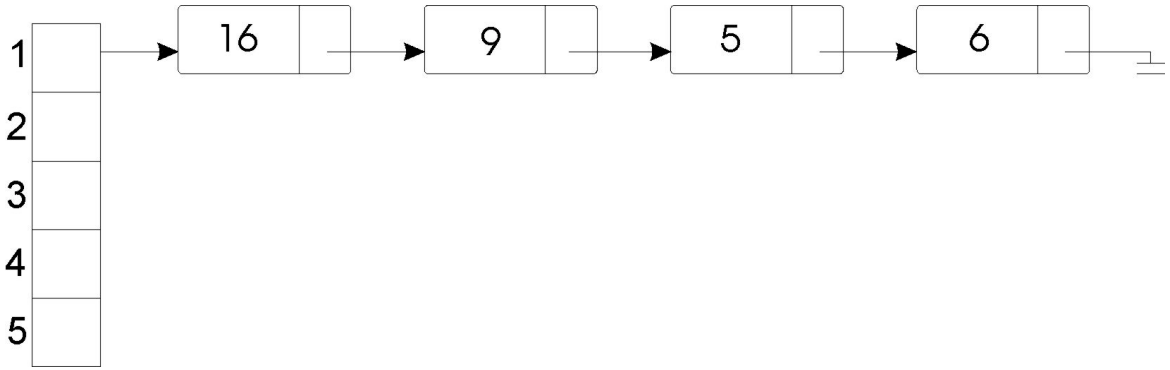


Frame Segment Tree - Speicherung

Speicherung ist abhängig vom jeweiligen Knoten:



Speicherung ist abhängig von den Objekten:



Automatische Indexierung

Es gibt drei Hauptschnitte in Filmen

a) Shot concatenation: Die Szenen werden aneinander gehängt

b) Spatial composition: Die Szenen werden ineinander übergeblendet

c) Chromatic composition: Entweder Ausblenden, oder Einblenden

Die automatische Indexierung kann den Einsatz des Menschen nicht übernehmen.

Audio-Retrieval

Es gibt vier Hauptsuchmethoden

1. Suche auf Metadaten
2. Suche auf akustischen Merkmalen
3. Suche mit Noten, bzw. Tonintervallen
4. Suche in gesprochenem Text

Suche auf Metadaten

Die Suche auf Metadaten erfolgt auf der Basis des Text-Retrieval

Zu den Audio-Daten wird eine Datenbank (z.B. CDDB) angelegt, die zusätzliche Informationen, wie Sängername, Gruppe, Erscheinungsjahr, etc. enthält.

Suche mit akustischen Merkmalen

- Suche erfolgt auf den „Rohdaten“ eines Audio-Signals

- Das Signal wird in einzelne Fenster a priori durch Eingabe einer bestimmten Fensterbreite oder a posteriori durch geeignete Mechanismen unterteilt.

- Man behandelt jedes Fenster einzeln und speichert die gewünschten Merkmale

Die bekanntesten Merkmale sind:

a) Intensität
b) Lautstärke
c) Tonhöhe
d) Helligkeit

Suche mit Noten

Die Suche erfolgt auf Basis des Taktes, Tempos oder der Noten

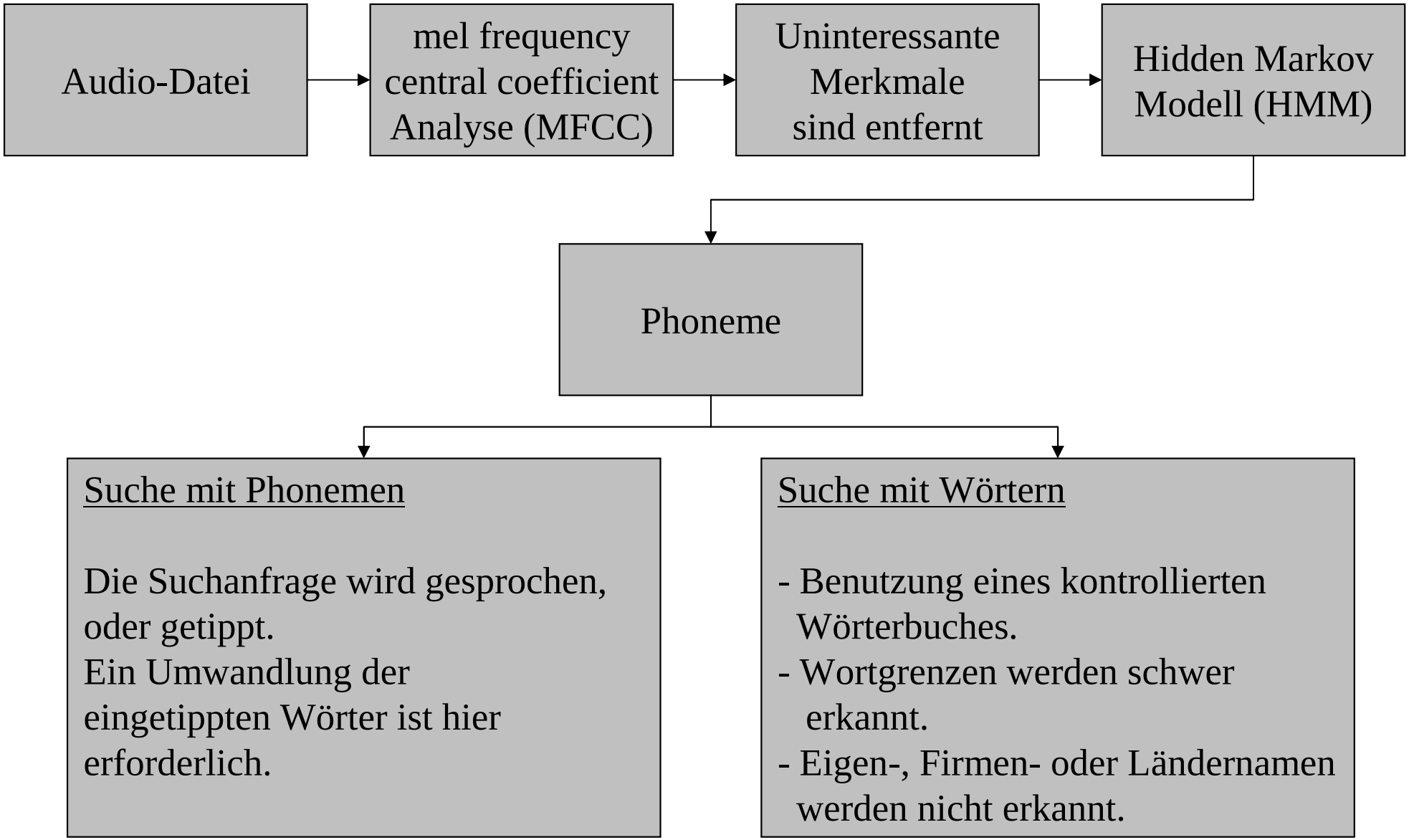
Erkennung der Noten aus dem Audiosignal:

Überlagerung vieler Instrumente führt zu Problemen
→ Verwendung von Audiodaten im MIDI-Format

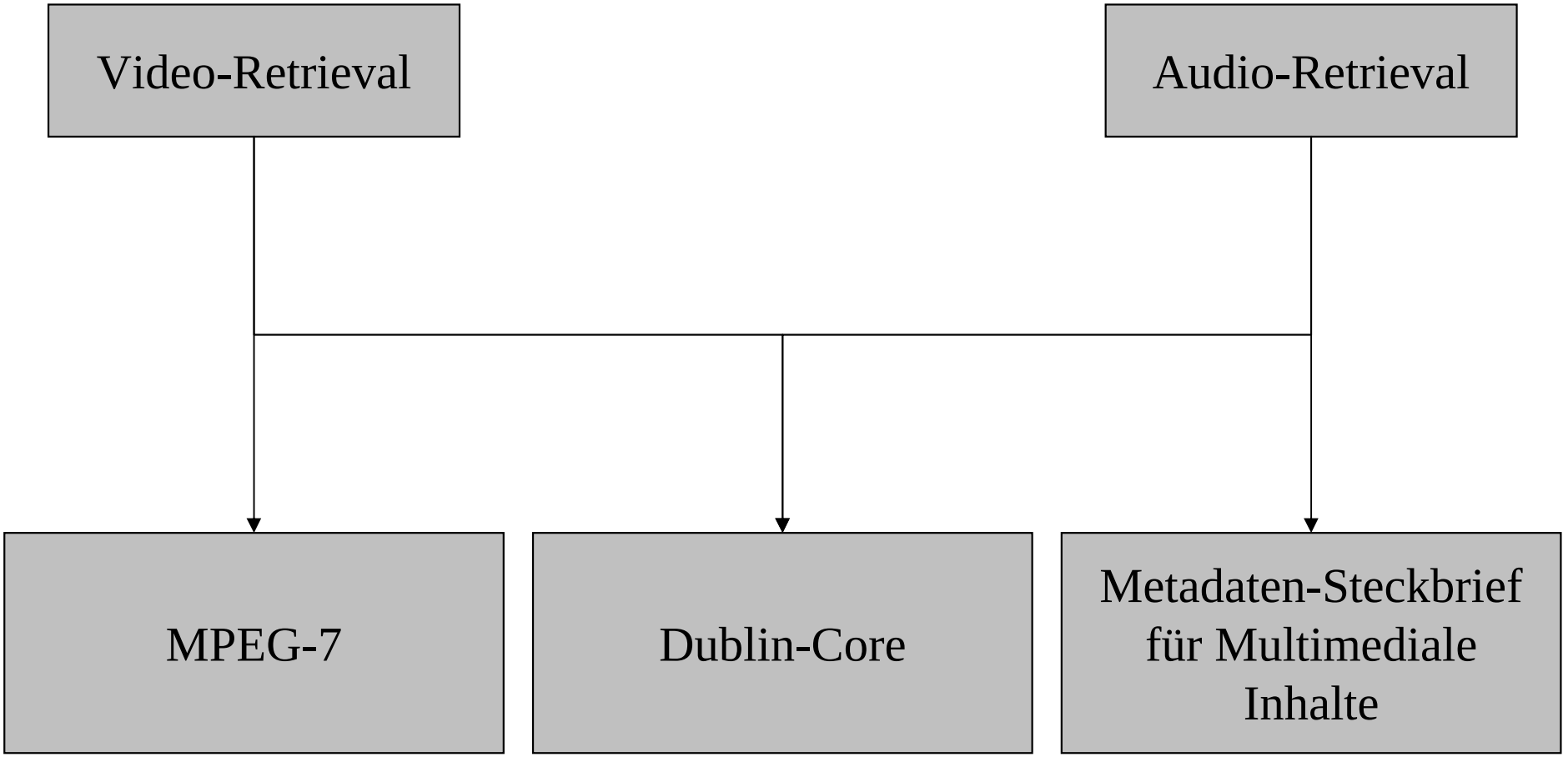
Search by Humming:

Die Melodie wird als Folge von „S“, „U“ und „D“ gespeichert
→ Unabhängig von der Tonhöhe

Suche in gesprochenem Text



Ausblick


Video-Retrieval
Audio-Retrieval
MPEG-7
Dublin-Core
Metadaten-Steckbrief für Multimediale Inhalte

ENDE