

# Shazam

Johannes Rolf, Lukas Stief

23. Januar 2018

# Übersicht

- 1 Einführung
  - Inhaltsbasierte Audiosuche
  - Ziele und Anforderungen an Shazam
- 2 Der Weg zum Fingerabdruck
  - Übersicht
  - Anpassungen am Signal
  - Constellation Map
- 3 Darstellung und Anfrage
  - Intuitiver Ansatz
  - Indizierung
  - Target Zones

# Inhaltsbasierte Audiosuche

- alle Anfragen mit zugrunde liegendem Sound
  - Aufnahmen
  - Audiodateien

# Inhaltsbasierte Audiosuche

- alle Anfragen mit zugrunde liegendem Sound
  - Aufnahmen
  - Audiodateien
- Unterschiede vor Allem in
  - der Länge der Audioanfrage und
  - der Spezifität des Anfrageziels

# Grundlegende Idee

- Berechnung so genannter Fingerabdrücke
  - bezeichnet spezielle Darstellung der Daten im Rechner

# Grundlegende Idee

- Berechnung so genannter Fingerabdrücke
  - bezeichnet spezielle Darstellung der Daten im Rechner
  - Genaues Aussehen von diesen ist stark vom eigentlichen Anwendungsfall abhängig.

# Grundlegende Idee

- Berechnung so genannter Fingerabdrücke
  - bezeichnet spezielle Darstellung der Daten im Rechner
  - Genaues Aussehen von diesen ist stark vom eigentlichen Anwendungsfall abhängig.
  - Unterstützung von gewünschten Anforderungen

# Anforderungen

**Ausgangssituation**

**Anforderungen**



# Anforderungen

## Ausgangssituation

Hintergrundgeräusche  
technisch bedingte Störgeräusche

## Anforderungen

# Anforderungen

## Ausgangssituation

Hintergrundgeräusche  
technisch bedingte Störgeräusche

## Anforderungen

⇒ robust

# Anforderungen

## Ausgangssituation

Hintergrundgeräusche  
technisch bedingte Störgeräusche  
unterschiedliche Aufnahmequalität

## Anforderungen

⇒ robust

# Anforderungen

## Ausgangssituation

Hintergrundgeräusche  
technisch bedingte Störgeräusche  
unterschiedliche Aufnahmequalität

## Anforderungen

⇒ robust  
⇒ formatunabhängig

# Anforderungen

## Ausgangssituation

Hintergrundgeräusche  
technisch bedingte Störgeräusche  
unterschiedliche Aufnahmequalität  
unabhängig vom Aufnahmezeitpunkt

## Anforderungen

⇒ robust  
⇒ formatunabhängig

# Anforderungen

## Ausgangssituation

Hintergrundgeräusche

technisch bedingte Störgeräusche

unterschiedliche Aufnahmequalität

unabhängig vom Aufnahmezeitpunkt

## Anforderungen

⇒ robust

⇒ formatunabhängig

⇒ zeitliche Lokalität

# Anforderungen

## Ausgangssituation

Hintergrundgeräusche  
technisch bedingte Störgeräusche  
unterschiedliche Aufnahmequalität  
unabhängig vom Aufnahmezeitpunkt  
i.A. schlechte Prozessorleistung

## Anforderungen

⇒ robust  
⇒ formatunabhängig  
⇒ zeitliche Lokalität

# Anforderungen

## Ausgangssituation

Hintergrundgeräusche  
technisch bedingte Störgeräusche  
unterschiedliche Aufnahmequalität  
unabhängig vom Aufnahmezeitpunkt  
i.A. schlechte Prozessorleistung

## Anforderungen

⇒ robust  
⇒ formatunabhängig  
⇒ zeitliche Lokalität  
⇒ wenig Rechenpower benötigt



# Anforderungen

## Ausgangssituation

Hintergrundgeräusche

technisch bedingte Störgeräusche

unterschiedliche Aufnahmequalität

unabhängig vom Aufnahmezeitpunkt

i.A. schlechte Prozessorleistung

keine langen Wartezeiten

## Anforderungen

⇒ robust

⇒ formatunabhängig

⇒ zeitliche Lokalität

⇒ wenig Rechenpower benötigt

# Anforderungen

## Ausgangssituation

Hintergrundgeräusche

technisch bedingte Störgeräusche

unterschiedliche Aufnahmequalität

unabhängig vom Aufnahmezeitpunkt

i.A. schlechte Prozessorleistung

keine langen Wartezeiten

## Anforderungen

⇒ robust

⇒ formatunabhängig

⇒ zeitliche Lokalität

⇒ wenig Rechenpower benötigt

⇒ hohe Geschwindigkeit

# Anforderungen

## Ausgangssituation

Hintergrundgeräusche  
technisch bedingte Störgeräusche  
unterschiedliche Aufnahmequalität  
unabhängig vom Aufnahmezeitpunkt  
i.A. schlechte Prozessorleistung  
keine langen Wartezeiten  
keine falschen Ergebnisse

## Anforderungen

⇒ robust  
⇒ formatunabhängig  
⇒ zeitliche Lokalität  
⇒ wenig Rechenpower benötigt  
⇒ hohe Geschwindigkeit

# Anforderungen

## Ausgangssituation

Hintergrundgeräusche  
technisch bedingte Störgeräusche  
unterschiedliche Aufnahmequalität  
unabhängig vom Aufnahmezeitpunkt  
i.A. schlechte Prozessorleistung  
keine langen Wartezeiten  
keine falschen Ergebnisse

## Anforderungen

⇒ robust  
⇒ formatunabhängig  
⇒ zeitliche Lokalität  
⇒ wenig Rechenpower benötigt  
⇒ hohe Geschwindigkeit  
⇒ hoch spezifisch

# Übersicht

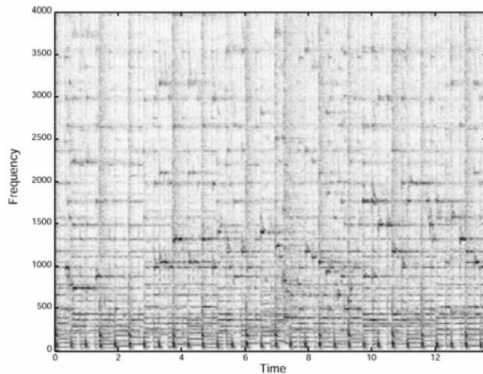


Abbildung: Spektrogramm

# Übersicht

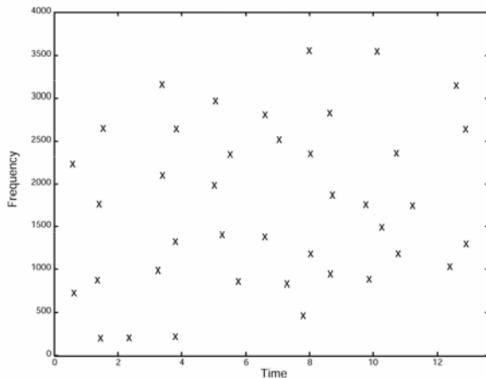


Abbildung: Constellation Map

# Übersicht

## Definitionen

$\mathcal{D} := \text{Dokument/Datenbank}$

$\mathcal{Q} := \text{Anfrage}$

## Definition: Constellation Map

Sei  $K \in \mathbb{N}$  die maximal mögliche Frequenz. Dann sind

$$\mathcal{C}(\mathcal{D}) \text{ bzw. } \mathcal{C}(\mathcal{Q}) \subset \mathbb{Z} \times [0 : K]$$

die Constellation Maps der Datenbank bzw. der Anfrage.

# Übersicht

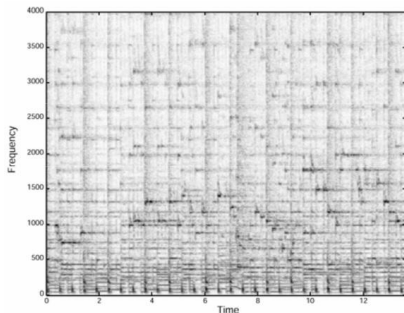


Abbildung: Spektrogramm

- Zu diesem Zeitpunkt:
  - Digitales Audiosignal



# Übersicht

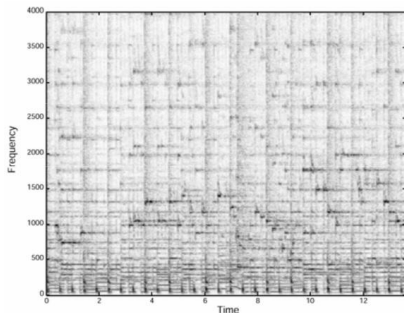


Abbildung: Spektrogramm

- Zu diesem Zeitpunkt:
  - Digitales Audiosignal
- Low pass Filter

# Übersicht

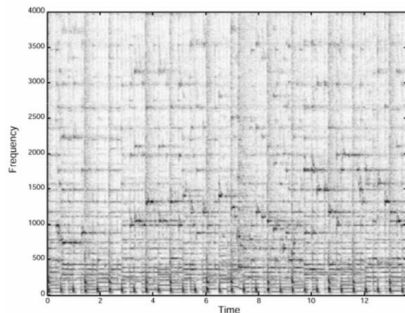


Abbildung: Spektrogramm

- Zu diesem Zeitpunkt:
  - Digitales Audiosignal
- Low pass Filter
- Downsampling

# Übersicht

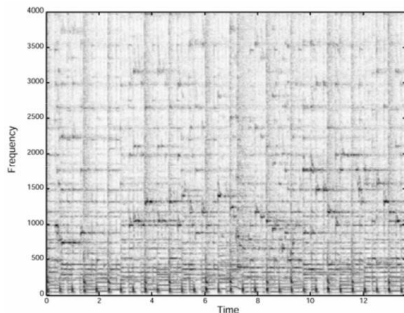


Abbildung: Spektrogramm

- Zu diesem Zeitpunkt:
  - Digitales Audiosignal
- Low pass Filter
- Downsampling
- Diskrete Fourier Transformation

# Low pass filter und Downsampling

- Frequenzen über 5 kHz werden aus dem Signal gefiltert

# Low pass filter und Downsampling

- Frequenzen über 5 kHz werden aus dem Signal gefiltert
- Das gefilterte Signal kann nun mit einer Samplingrate von 11,025 kHz gesamplet werden

# Low pass filter und Downsampling

- Frequenzen über 5 kHz werden aus dem Signal gefiltert
- Das gefilterte Signal kann nun mit einer Samplingrate von 11,025 kHz gesampelt werden
- Dabei entsteht kein Aliasing

# Low pass filter und Downsampling

- Frequenzen über 5 kHz werden aus dem Signal gefiltert
- Das gefilterte Signal kann nun mit einer Samplingrate von 11,025 kHz gesampelt werden
- Dabei entsteht kein Aliasing
- Grund: Diskrete Fourier Transformation benötigt weniger Zeit ohne Genauigkeit einzubüßen

# Erstellen des Spektrogramms

- Um die Frequenzen des digitalen Signals zu erhalten wird die Diskrete Fourier Transformation angewandt



# Erstellen des Spektrogramms

- Um die Frequenzen des digitalen Signals zu erhalten wird die Diskrete Fourier Transformation angewandt
- Wir wählen die Spezifikationen so, dass in 0.1 Sekunden Abschnitten unterschieden werden kann

## Zu diesem Zeitpunkt

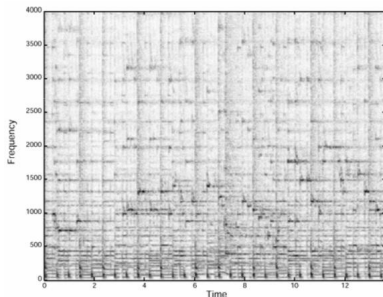


Abbildung: Spektrogramm

- Zu diesem Zeitpunkt haben wir ein Spektrogramm mit folgenden Eigenschaften:

## Zu diesem Zeitpunkt

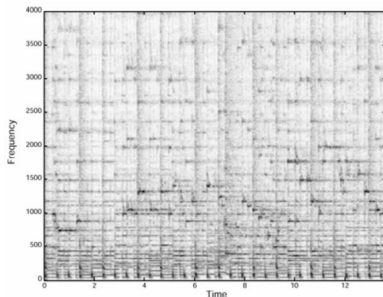


Abbildung: Spektrogramm

- Zu diesem Zeitpunkt haben wir ein Spektrogramm mit folgenden Eigenschaften:
  - Zeitintervalle von 0.1 Sekunden

## Zu diesem Zeitpunkt

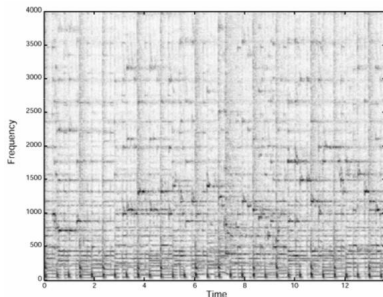


Abbildung: Spektrogramm

- Zu diesem Zeitpunkt haben wir ein Spektrogramm mit folgenden Eigenschaften:
  - Zeitintervalle von 0.1 Sekunden
  - 0 Hz bis 5 kHz

## Zu diesem Zeitpunkt

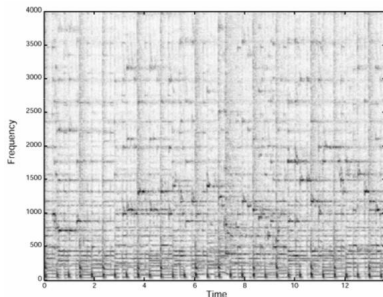


Abbildung: Spektrogramm

- Zu diesem Zeitpunkt haben wir ein Spektrogramm mit folgenden Eigenschaften:
  - Zeitintervalle von 0.1 Sekunden
  - 0 Hz bis 5 kHz
  - 512 mögliche Frequenzbereiche

# Zu diesem Zeitpunkt

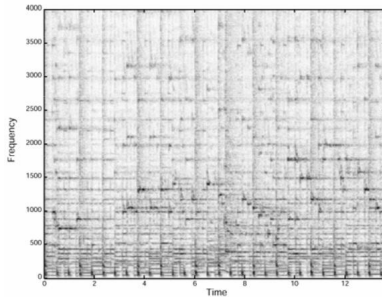


Abbildung: Zu diesem Zeitpunkt

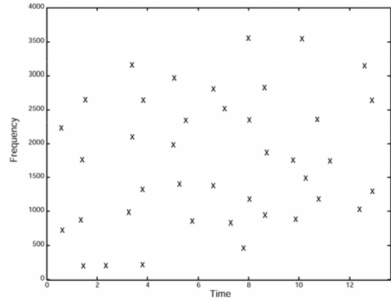
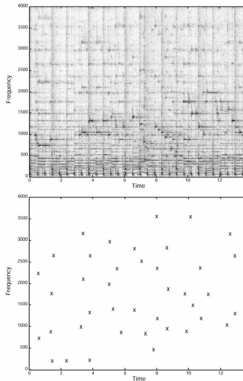


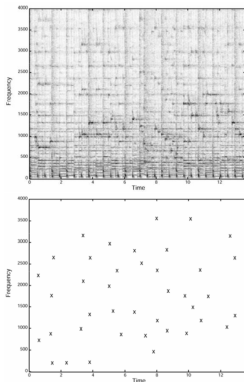
Abbildung: Das nächste Ziel

# Übersicht



- Nur die lautesten Frequenzen sollen gespeichert werden

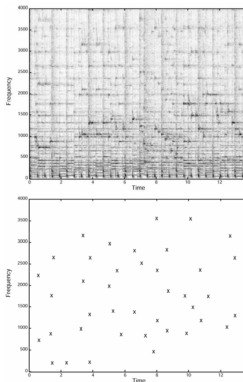
# Übersicht



- Nur die lautesten Frequenzen sollen gespeichert werden
  - Oft werden tiefe Frequenzen künstlich verstärkt

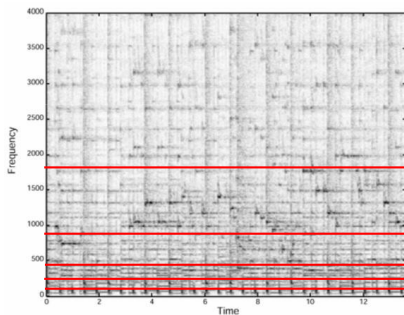


# Übersicht



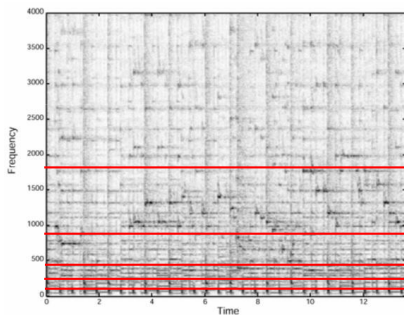
- Nur die lautesten Frequenzen sollen gespeichert werden
  - Oft werden tiefe Frequenzen künstlich verstärkt
  - Deshalb wird das Spektrogramm in sechs (logarithmische) Bänder aufgeteilt.

# Aufteilung in Bänder



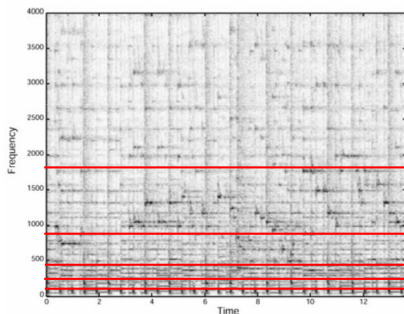
- In jeder zehntel Sekunde wird der lauteste Frequenzbereich betrachtet

# Aufteilung in Bänder



- In jeder zehntel Sekunde wird der lauteste Frequenzbereich betrachtet
- Der Durchschnitt dieser sechs Werte wird berechnet

# Aufteilung in Bänder



- In jeder zehntel Sekunde wird der lauteste Frequenzbereich betrachtet
- Der Durchschnitt dieser sechs Werte wird berechnet
- Behalte Frequenzen die (multipliziert mit einem Koeffizienten) über dem Durchschnitt liegen

# Constellation Map

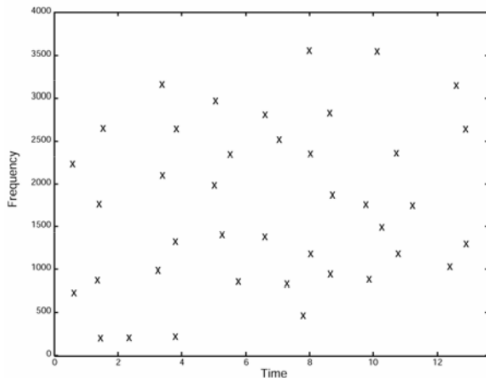


Abbildung: Constellation Map

# Inhaltsverzeichnis

- 1 Einführung
  - Inhaltsbasierte Audiosuche
  - Ziele und Anforderungen an Shazam
- 2 Der Weg zum Fingerabdruck
  - Übersicht
  - Anpassungen am Signal
  - Constellation Map
- 3 Darstellung und Anfrage
  - Intuitiver Ansatz
  - Indizierung
  - Target Zones

# Anfrage

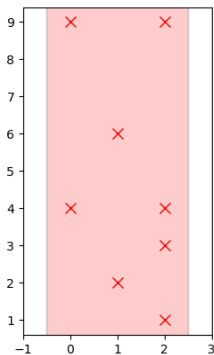


Abbildung: Anfrage

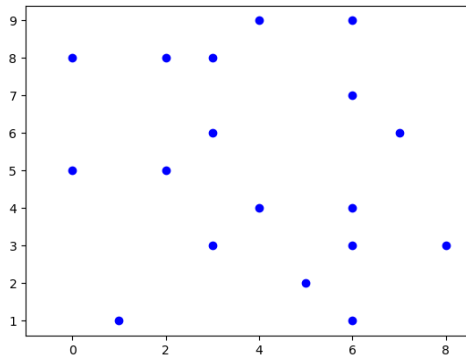
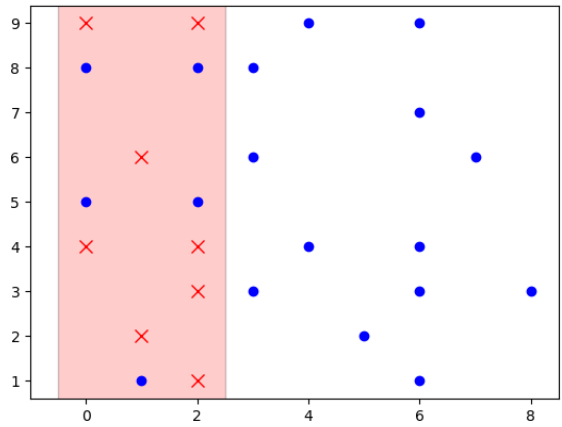


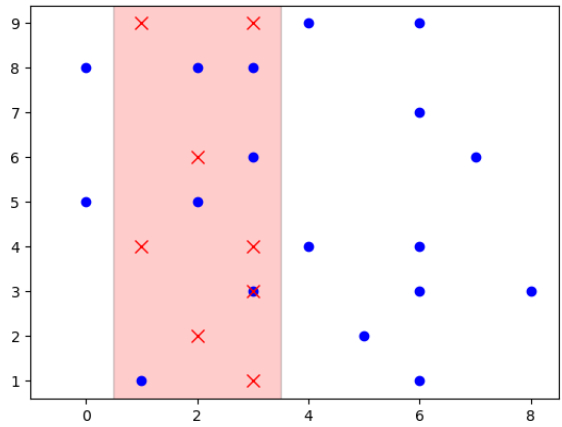
Abbildung: Datenbank

# Intuitiver Ansatz

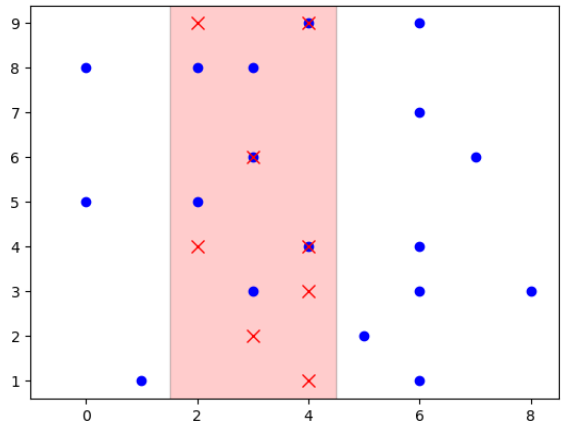




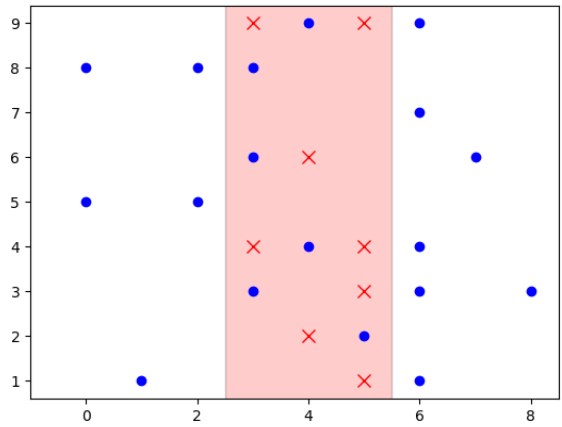
# Intuitiver Ansatz



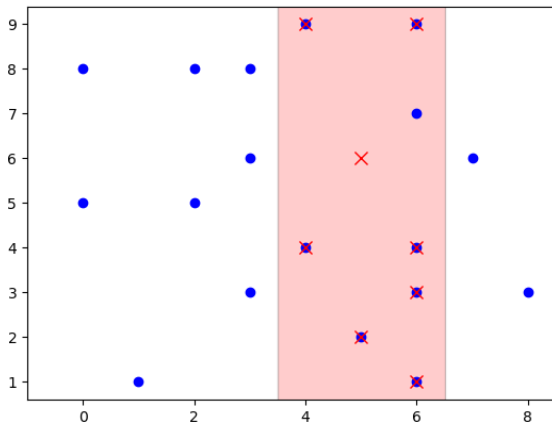
# Intuitiver Ansatz



# Intuitiver Ansatz



# Intuitiver Ansatz



# Laufzeit

## Definition: Constellation Map

Sei  $K \in \mathbb{N}$  die maximal mögliche Frequenz. Dann sind

$$\mathcal{C}(\mathcal{D}) \text{ bzw. } \mathcal{C}(\mathcal{Q}) \subset \mathbb{Z} \times [0 : K]$$

die Constellation Maps der Datenbank bzw. der Anfrage.

## Laufzeit: Intuitiver Ansatz

# Laufzeit

## Definition: Constellation Map

Sei  $K \in \mathbb{N}$  die maximal mögliche Frequenz. Dann sind

$$\mathcal{C}(\mathcal{D}) \text{ bzw. } \mathcal{C}(\mathcal{Q}) \subset \mathbb{Z} \times [0 : K]$$

die Constellation Maps der Datenbank bzw. der Anfrage.

## Laufzeit: Intuitiver Ansatz

$$\text{Linear in } |\mathcal{C}(\mathcal{D})| * |\mathcal{C}(\mathcal{Q})|$$

# Laufzeit

## Definition: Constellation Map

Sei  $K \in \mathbb{N}$  die maximal mögliche Frequenz. Dann sind

$$\mathcal{C}(\mathcal{D}) \text{ bzw. } \mathcal{C}(\mathcal{Q}) \subset \mathbb{Z} \times [0 : K]$$

die Constellation Maps der Datenbank bzw. der Anfrage.

## Laufzeit: Intuitiver Ansatz

Linear in  $|\mathcal{C}(\mathcal{D})| * |\mathcal{C}(\mathcal{Q})|$   
 $\Rightarrow$  deutlich zu langsam!

# Exkurs

## Stichwort- und Namenverzeichnis

- Ablage
  - differenzieren **146**
  - eines Vorgangs **26f.**
  - hängend **20**
  - liegend **18**
  - stehend **19**
- Ablagestruktur **129**
- Ablenkung **50**
- Aktenordner **33, 38, 46, 53, 71**
- Aktensystemplan **50, 57, 206ff., 215**
- Akten tasche **44f.**
- Aktentrennung **43**
- Aktionsmappen **89**
- »Aktive Projekte« **149**
- Allen, David* **148**
- Außendienstmitarbeiter **168**
- Beierle, Beate* **166f.**
- Beschriftungsgerät **97**
- Brother **97, 175**
- Brutto-Netto-Verhältnis **35f.**
- Bürofachhandel **68**
- Bürofrust **53**
- Büro, papierloses **14f.**
- Bush, Vannevar* **15**
- CD-Tasche **116**
- Chaos **7, 55, 179**
- Classi* **9**
- Clear-Desk-Policy **124**
- Einstellmappen **19, 67**
- Ergänzungsmappen **81**
- Fächer mappen **82, 193, 198**
- Farbcodierung **58**
- Farbskala **95, 192, 209**
- Fehler **59**
- Fensterreiter **94**
- Flexibilität **40, 41, 43, 45f., 156**
- Folienreiter **94, 97, 139**
- Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA* **31**
- Fruststufen **53f.**



# Idee

Analog zum Stichwortverzeichnis:

- Einer Frequenz wird eine Liste an Zeitpunkten zugeordnet, an denen diese Frequenz auftritt.

# Idee

Analog zum Stichwortverzeichnis:

- Einer Frequenz wird eine Liste an Zeitpunkten zugeordnet, an denen diese Frequenz auftritt.
- Verwende Hash-Listen für effizienten Zugriff auf diese Listen.

# Feature Representation

## Definition: Constellation Map

Sei  $K \in \mathbb{N}$  die maximal mögliche Frequenz. Dann sind

$$\mathcal{C}(\mathcal{D}) \text{ bzw. } \mathcal{C}(\mathcal{Q}) \subset \mathbb{Z} \times [0 : K]$$

die Constellation Maps der Datenbank bzw. der Anfrage.

## Definition: Feature Representation

Sei  $\mathcal{H}$  eine endliche Menge möglicher Hash-Werte. Dann sind

$$\mathcal{F}(\mathcal{D}) \text{ bzw. } \mathcal{F}(\mathcal{Q}) \subset \mathbb{Z} \times \mathcal{H}$$

die Feature Representations der Datenbank bzw. der Anfrage.

# Hash-Listen

## Definition: Feature Representation

Sei  $\mathcal{H}$  eine endliche Menge möglicher Hash-Werte. Dann sind

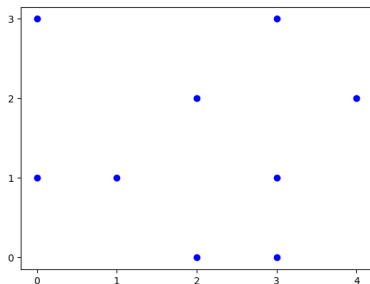
$$\mathcal{F}(\mathcal{D}) \text{ bzw. } \mathcal{F}(\mathcal{Q}) \subset \mathbb{Z} \times \mathcal{H}$$

die Feature Representations der Datenbank bzw. der Anfrage.

## Definition: Hash-Liste

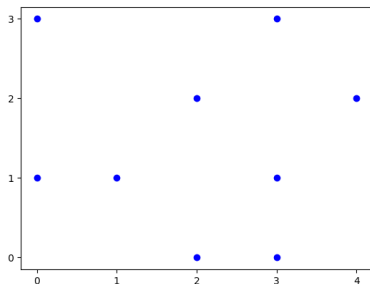
Für einen Hash-Wert  $h \in \mathcal{H}$  sei  $\mathcal{L}(h)$  eine geordnete Liste mit allen Zeitpunkten  $t \in \mathbb{Z}$  mit  $(t, h) \in \mathcal{F}(\mathcal{D})$

# Beispiel



$$\mathcal{H} = \{0, 1, 2, 3\}$$

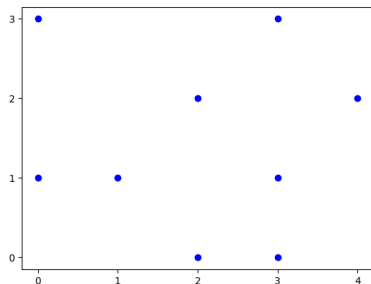
# Beispiel



$$\mathcal{H} = \{0, 1, 2, 3\}$$

$$\mathcal{L}(0) = (2, 3)$$

# Beispiel

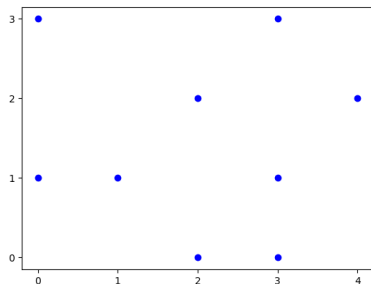


$$\mathcal{H} = \{0, 1, 2, 3\}$$

$$\mathcal{L}(0) = (2, 3)$$

$$\mathcal{L}(1) = (0, 1, 3)$$

# Beispiel



$$\mathcal{H} = \{0, 1, 2, 3\}$$

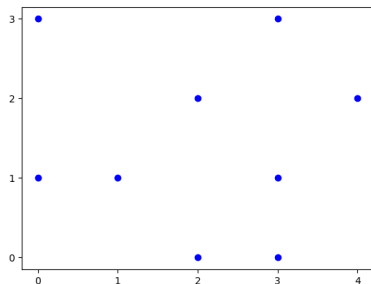
$$\mathcal{L}(0) = (2, 3)$$

$$\mathcal{L}(1) = (0, 1, 3)$$

$$\mathcal{L}(2) = (2, 4)$$



# Beispiel



$$\mathcal{H} = \{0, 1, 2, 3\}$$

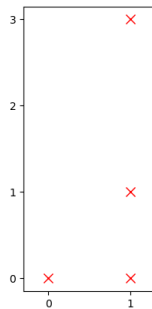
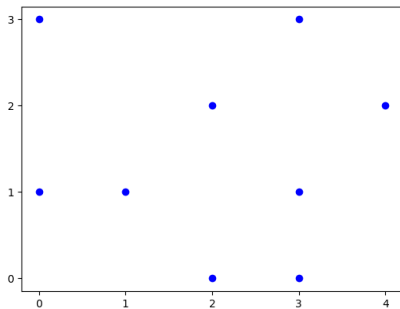
$$\mathcal{L}(0) = (2, 3)$$

$$\mathcal{L}(1) = (0, 1, 3)$$

$$\mathcal{L}(2) = (2, 4)$$

$$\mathcal{L}(3) = (0, 3)$$

# Anfrage



# Matching-Function

Query (t,h)	$\mathcal{L}(h)$	Indicator functions						
		...	-1	0	1	2	3	4
(0,0)	(2,3)	0	0	0	0	1	1	0
(1,0)	(2,3)	0	0	0	1	1	0	0
(1,1)	(0,1,3)	0	0	1	0	1	0	1
(1,3)	(0,3)	0	1	0	0	1	0	0
<b>Matching function</b>		<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>4</b>	<b>1</b>	<b>1</b>

# Laufzeit

Annahme:

- Alle Hash-Werte in  $\mathcal{H}$  sind über der Datenbank gleichverteilt

# Laufzeit

Annahme:

- Alle Hash-Werte in  $\mathcal{H}$  sind über der Datenbank gleichverteilt

Folgerung

$$|\mathcal{L}(h)| \approx \frac{|\mathcal{F}(\mathcal{D})|}{|\mathcal{H}|}, \forall h \in \mathcal{H}$$

# Laufzeit

Annahme:

- Alle Hash-Werte in  $\mathcal{H}$  sind über der Datenbank gleichverteilt

Folgerung

$$|\mathcal{L}(h)| \approx \frac{|\mathcal{F}(\mathcal{D})|}{|\mathcal{H}|}, \forall h \in \mathcal{H}$$

Zur Abfrage benötigen wir nur die Elemente aus  $\mathcal{F}(Q)$ . Diese müssen nur angesprochen, verschoben und verglichen werden.  
 $\Rightarrow$  Alles Funktionen, die linear in  $|\mathcal{F}(Q)|$  sind.

# Laufzeit

Annahme:

- Alle Hash-Werte in  $\mathcal{H}$  sind über der Datenbank gleichverteilt

Folgerung

$$|\mathcal{L}(h)| \approx \frac{|\mathcal{F}(\mathcal{D})|}{|\mathcal{H}|}, \forall h \in \mathcal{H}$$

Zur Abfrage benötigen wir nur die Elemente aus  $\mathcal{F}(Q)$ . Diese müssen nur angesprochen, verschoben und verglichen werden.  
 $\Rightarrow$  Alles Funktionen, die linear in  $|\mathcal{F}(Q)|$  sind.

Laufzeit

$$\text{Linear in } |\mathcal{F}(Q)| * \frac{|\mathcal{F}(\mathcal{D})|}{|\mathcal{H}|}$$

# Laufzeit

Annahme:

- Alle Hash-Werte in  $\mathcal{H}$  sind über der Datenbank gleichverteilt

Folgerung

$$|\mathcal{L}(h)| \approx \frac{|\mathcal{F}(\mathcal{D})|}{|\mathcal{H}|}, \forall h \in \mathcal{H}$$

Zur Abfrage benötigen wir nur die Elemente aus  $\mathcal{F}(Q)$ . Diese müssen nur angesprochen, verschoben und verglichen werden.  
 $\Rightarrow$  Alles Funktionen, die linear in  $|\mathcal{F}(Q)|$  sind.

Laufzeit

$$\text{Linear in } |\mathcal{F}(Q)| * \frac{|\mathcal{F}(\mathcal{D})|}{|\mathcal{H}|}$$

$\Rightarrow$  Verbesserung um Faktor  $|\mathcal{H}|$



# Überblick

## Wiederholung: Feature Representation

Sei  $\mathcal{H}$  eine endliche Menge möglicher Hash-Werte. Dann sind

$$\mathcal{F}(\mathcal{D}) \text{ bzw. } \mathcal{F}(\mathcal{Q}) \subset \mathbb{Z} \times \mathcal{H}$$

die Feature Representations der Datenbank bzw. der Anfrage.

- Ziel: Vergrößere  $|\mathcal{H}|$  um Laufzeit zu verbessern

# Überblick

## Wiederholung: Feature Representation

Sei  $\mathcal{H}$  eine endliche Menge möglicher Hash-Werte. Dann sind

$$\mathcal{F}(\mathcal{D}) \text{ bzw. } \mathcal{F}(\mathcal{Q}) \subset \mathbb{Z} \times \mathcal{H}$$

die Feature Representations der Datenbank bzw. der Anfrage.

- Ziel: Vergrößere  $|\mathcal{H}|$  um Laufzeit zu verbessern
- Idee:
  - Anstatt jeden Punkt der Constellation Map einzeln zu betrachten, betrachte mehrere Punkte zeitgleich

# Überblick

## Wiederholung: Feature Representation

Sei  $\mathcal{H}$  eine endliche Menge möglicher Hash-Werte. Dann sind

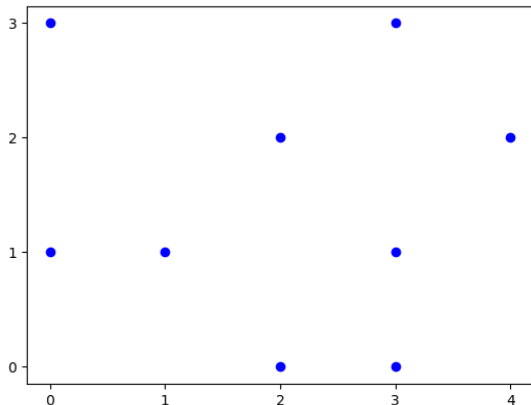
$$\mathcal{F}(\mathcal{D}) \text{ bzw. } \mathcal{F}(\mathcal{Q}) \subset \mathbb{Z} \times \mathcal{H}$$

die Feature Representations der Datenbank bzw. der Anfrage.

- Ziel: Vergrößere  $|\mathcal{H}|$  um Laufzeit zu verbessern
- Idee:
  - Anstatt jeden Punkt der Constellation Map einzeln zu betrachten, betrachte mehrere Punkte zeitgleich
  - Dazu werden die Punkte geordnet und in sogenannte Target Zones aufgeteilt

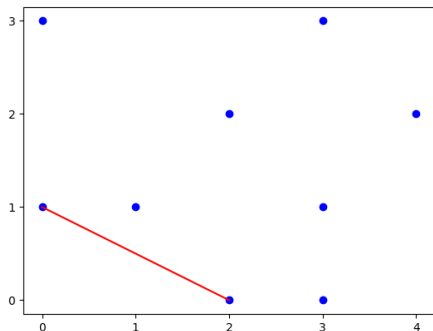
# Beispiel

In  $\mathcal{F}(\mathcal{D})$  nun Elemente der Form  $(n_0, (f_0, f_1, n_1 - n_0))$



# Beispiel

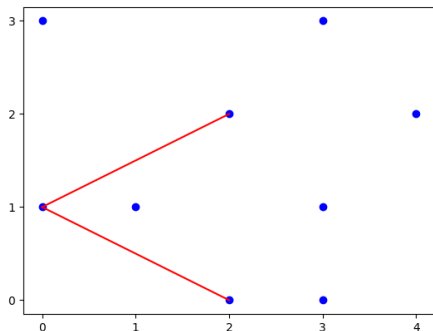
In  $\mathcal{F}(\mathcal{D})$  nun Elemente der Form  $(n_0, (f_0, f_1, n_1 - n_0))$



$(0, 1)$  wird zu  
 $(0, (1, 0, 2))$

# Beispiel

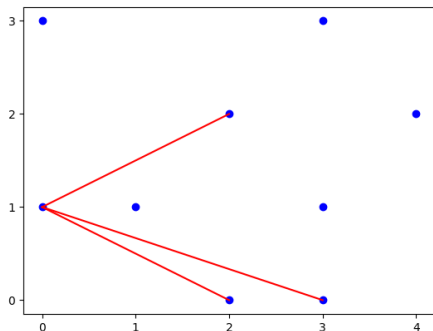
In  $\mathcal{F}(\mathcal{D})$  nun Elemente der Form  $(n_0, (f_0, f_1, n_1 - n_0))$



$(0, 1)$  wird zu  
 $(0, (1, 0, 2))$  und  
 $(0, (1, 2, 2))$

# Beispiel

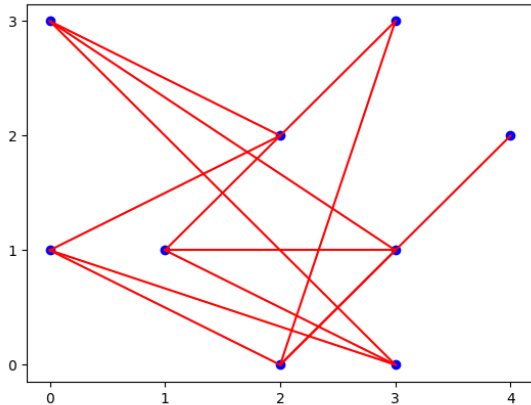
In  $\mathcal{F}(\mathcal{D})$  nun Elemente der Form  $(n_0, (f_0, f_1, n_1 - n_0))$



$(0, 1)$  wird zu

$(0, (1, 0, 2))$  und  
 $(0, (1, 2, 2))$  und  
 $(0, (1, 0, 3))$

# Beispiel





# Target Zone

Aufbau von Target Zones:

- Eine Menge von Punkten in der Constellation Map
- Diese Punkte stehen in Relation zu einem Ankerpunkt
- Ankerpunkt hat einen Abstand zur Target Zone

# Laufzeit

Annahmen:

- konstanter Faktor  $F \in \mathbb{N}$  für die Anzahl der Punkte in einer Target Zone

$$\Rightarrow F * |\mathcal{F}(\mathcal{D})_{single}| = |\mathcal{F}(\mathcal{D})_{triple}|$$

$$\Rightarrow F * |\mathcal{F}(\mathcal{Q})_{single}| = |\mathcal{F}(\mathcal{Q})_{triple}|$$

# Laufzeit

Annahmen:

- konstanter Faktor  $F \in \mathbb{N}$  für die Anzahl der Punkte in einer Target Zone

$$\Rightarrow F * |\mathcal{F}(\mathcal{D})_{single}| = |\mathcal{F}(\mathcal{D})_{triple}|$$

$$\Rightarrow F * |\mathcal{F}(\mathcal{Q})_{single}| = |\mathcal{F}(\mathcal{Q})_{triple}|$$

- Frequenzen und Zeitpunkte werden jeweils durch  $B$  Bits kodiert

$$\Rightarrow 2^B \quad \text{Hashes für } f_0$$

$$\Rightarrow 2^{B+B+B} \quad \text{Hashes für } (f_0, f_1, t_1 - t_0)$$

# Laufzeit

## Erinnerung: Laufzeit

Linear in  $|\mathcal{F}(Q)| * \frac{|\mathcal{F}(\mathcal{D})|}{|\mathcal{H}|}$

- Version ohne Target Zones:

$$|\mathcal{F}(Q)_{single}| * \frac{|\mathcal{F}(\mathcal{D})_{single}|}{2^B} =: T_{single}$$

# Laufzeit

- Version mit Target Zones:

$$|\mathcal{F}(Q)_{triple}| * \frac{|\mathcal{F}(D)_{triple}|}{2^{B+B+B}}$$

# Laufzeit

- Version mit Target Zones:

$$|\mathcal{F}(Q)_{triple}| * \frac{|\mathcal{F}(D)_{triple}|}{2^{B+B+B}} = F * |\mathcal{F}(Q)_{single}| * \frac{F * |\mathcal{F}(D)_{single}|}{2^{B+B+B}}$$

# Laufzeit

- Version mit Target Zones:

$$\begin{aligned} |\mathcal{F}(Q)_{triple}| * \frac{|\mathcal{F}(\mathcal{D})_{triple}|}{2^{B+B+B}} &= F * |\mathcal{F}(Q)_{single}| * \frac{F * |\mathcal{F}(\mathcal{D})_{single}|}{2^{B+B+B}} \\ &= \frac{F^2}{2^{B+B}} * |\mathcal{F}(Q)_{single}| * \frac{|\mathcal{F}(\mathcal{D})_{single}|}{2^B} \end{aligned}$$

# Laufzeit

- Version mit Target Zones:

$$\begin{aligned} |\mathcal{F}(Q)_{triple}| * \frac{|\mathcal{F}(\mathcal{D})_{triple}|}{2^{B+B+B}} &= F * |\mathcal{F}(Q)_{single}| * \frac{F * |\mathcal{F}(\mathcal{D})_{single}|}{2^{B+B+B}} \\ &= \frac{F^2}{2^{B+B}} * |\mathcal{F}(Q)_{single}| * \frac{|\mathcal{F}(\mathcal{D})_{single}|}{2^B} \\ &= \frac{F^2}{2^{B+B}} * T_{single} \end{aligned}$$



# Laufzeit

- Version mit Target Zones:

$$\begin{aligned} |\mathcal{F}(Q)_{triple}| * \frac{|\mathcal{F}(\mathcal{D})_{triple}|}{2^{B+B+B}} &= F * |\mathcal{F}(Q)_{single}| * \frac{F * |\mathcal{F}(\mathcal{D})_{single}|}{2^{B+B+B}} \\ &= \frac{F^2}{2^{B+B}} * |\mathcal{F}(Q)_{single}| * \frac{|\mathcal{F}(\mathcal{D})_{single}|}{2^B} \\ &= \frac{F^2}{2^{B+B}} * T_{single} \end{aligned}$$

## Laufzeit

Laufzeitverbesserung um  $\frac{2^{B+B}}{F^2}$

# Beispiel

## Laufzeit

Laufzeitverbesserung um  $\frac{2^{B+B}}{F^2}$

- $B := 10$
- $F := 10$

# Beispiel

## Laufzeit

Laufzeitverbesserung um  $\frac{2^{B+B}}{F^2}$

- $B := 10$
- $F := 10$

$\Rightarrow$  Laufzeitverbesserung um  $\frac{2^{20}}{100} \approx 10000$