- Passive digitale Fingerabdrücke

Passive digitale Fingerabdrücke

- Methode, um den Inhalt von Audiodateien im Vergleich zu identifizieren
- Synonyme:
  - Robust Audio Hash
  - Audio ID
  - Audio Fingerprinting

- Beachten:
  - Passive Fingerabdrücke = Robuster Hash
  - Aktive Fingerabdrücke = Kunden-Wasserzeichen

Passive digitale Fingerabdrücke

- Wozu?
  - Broadcast Monitoring
  - Identifizierung von geschütztem Material bei CD-Duplikation
  - Intelligente Geräte, die auf Medien reagieren
  - Filtern von Medien in Netzwerken (z.B. P2P)
  - Erkennen von Manipulationen an digitalen Medien
  - Synchronisation von Wasserzeichen-Markierungen

Wozu?

„Name that tune ...“

1. Musikstück hören
2. Über Handy weiterleiten
3. Zentraler Rechner sucht passendes Musikstück
4. SMS mit Name zurück

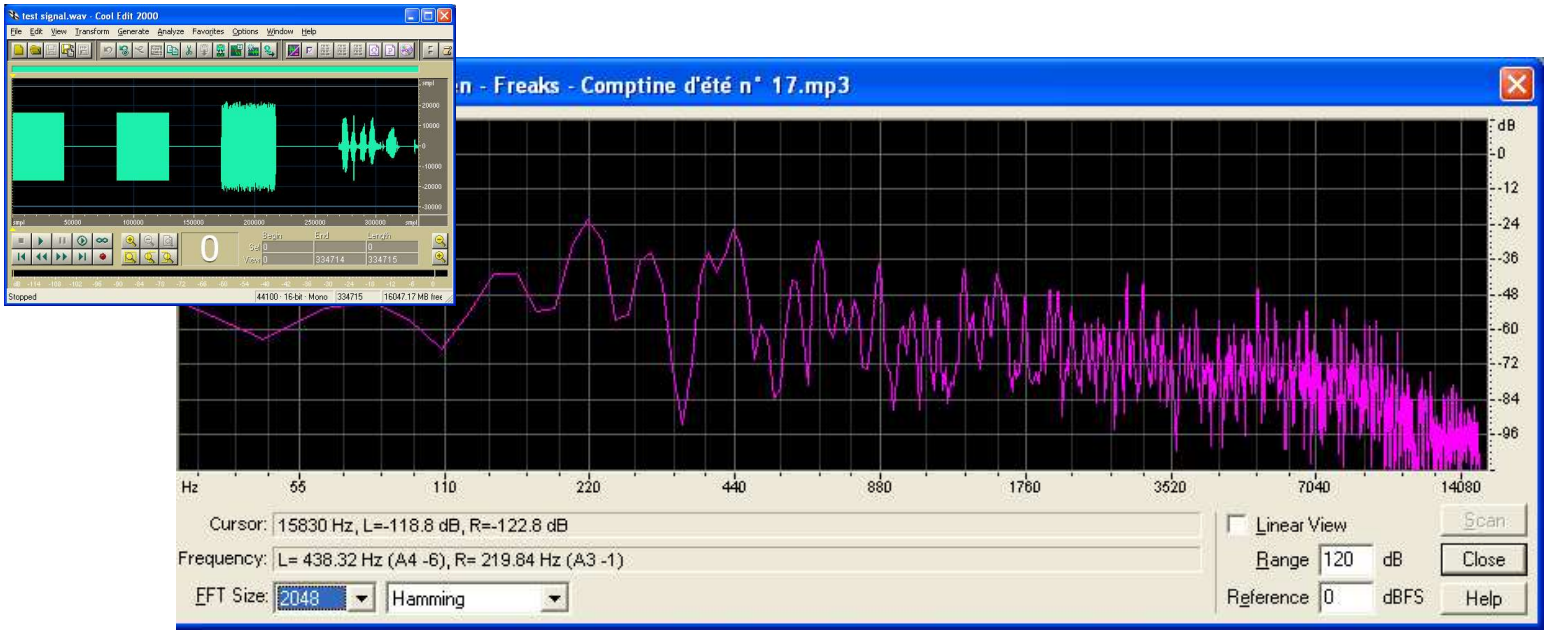
Name that tune

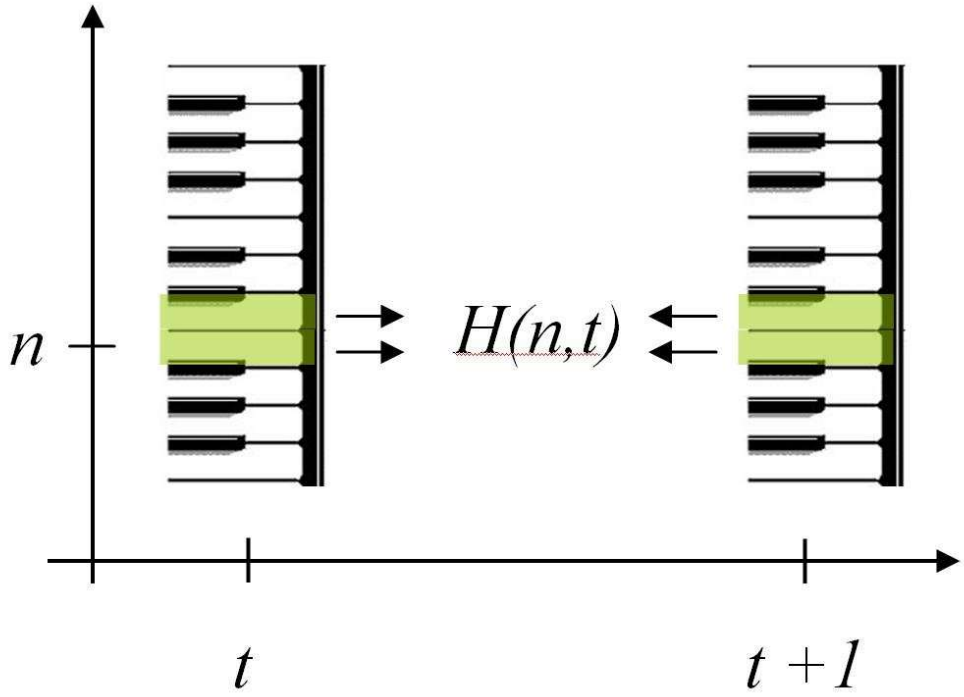
QUELLE: http://www.research.phillips.com/

Beispiel: O2 Music Spy Dienst

Vergleichbare Dienste von
- E-Plus: („Snap a Song“) oder „m2any”
- Geschäftsmodell: Marketinginstrument oder Dienstleistung

- Beispiel-1: Fingerprinting nach [Haitsma, Kalker] (Phillips)
- Prinzip
  - Gewonnen aus 33 Frequenzbändern zwischen 300 Hz und 3000 Hz
  - Breite der Bänder: ca. eine Note (Halbton); logarithmische Frequenzskala
  - 32 bit Hash für jedes Frame von 0,4 Sekunden

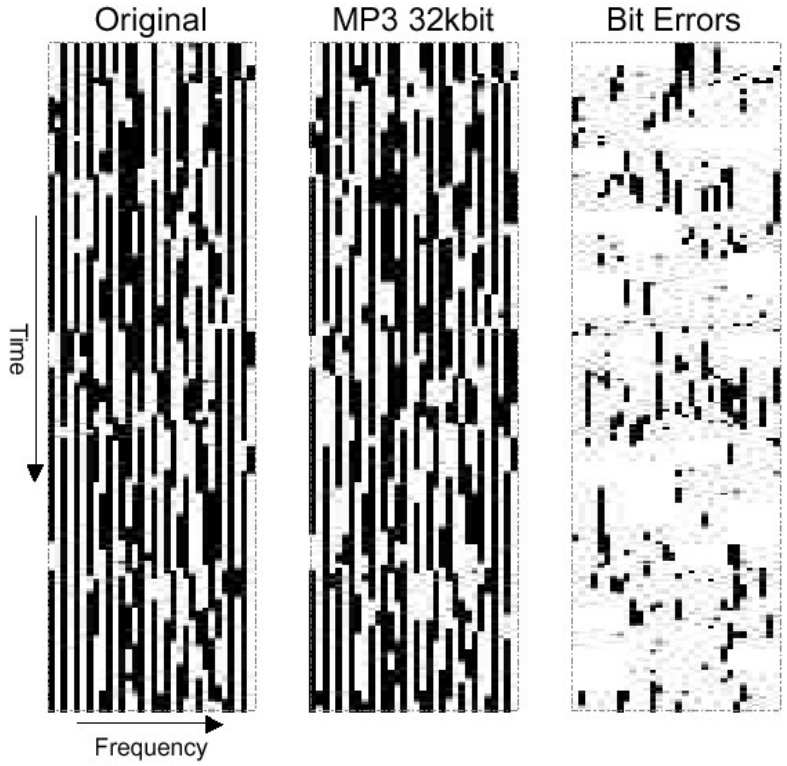




$$H(n,m) = \begin{cases} 1 & \text{if } EB(n,m) - EB(n,m+1) - \big(EB(n-1,m) - EB(n-1,m+1)\big) > 0 \\ 0 & \text{if } EB(n,m) - EB(n,m+1) - \big(EB(n-1,m) - EB(n-1,m+1)\big) \le 0 \end{cases}$$

•Quelle: Robust Audio Hashing for Content Identification, Jaap Haitsma, Ton Kalker and Job Oostveen, Philips Research

- Quelle: Robust Audio Hashing for Content Identification, Jaap Haitsma, Ton Kalker and Job Oostveen, Philips Research



„A robust audio hash is a function that associates to every basic time-unit of audio content a short semi-unique bit-sequence that is continuous with respect to content similarity as perceived by the HAS.“

- Quelle: Robust Audio Hashing for Content Identification, Jaap Haitsma, Ton Kalker and Job Oostveen, Philips Research

Processing	Orff	Sinead	Texas	ACDC
MP3@128Kbps	0.078	0.086	0.085	0.084
MP3@32Kbps	0.177	0.106	0.098	0.136
Real@20Kbps	0.160	0.138	0.160	0.209
All-pass filtering	0.019	0.016	0.017	0.027
Amp. Compr.	0.053	0.075	0.113	0.073
Equalization	0.049	0.044	0.065	0.062
Echo Addition	0.157	0.144	0.140	0.144
Band Pass Filter	0.028	0.026	0.024	0.038
Resampling	0.000	0.000	0.000	0.000
D/A A/D	0.088	0.061	0.112	0.074

Fehlerraten nach „Angriffen“.

Passive Fingerabdrücke

- Beispiel-2: Fraunhofer AudioID:
- Ansatz: MPEG-7 Deskriptoren
  - MPEG-7: Beschreibungssprache für Metadaten von Multimedia-Informationen; kein Kompressionsformat!
  - Audio Features (Low level descriptors LLD)
  - Unterteilung der Audiosignals in Frames:
  - Gruppierung des Klangspektrums in ¼ Oktaven (3 Halbtöne)
  - Berechnung der spektralen Glattheit (SpectrumFlatness LLD)
  - Tonales Spektrum? Rauschartig??

QUELLE: http://www.imk.fraunhofer.de/sixcms/media.php/208/hellmuth_audioid.pdf

- SpectrumFlatness: Quotient des geometrischen Mittels *g* und arithmetischen Mittels *m* der Energie in den Frequenzändern

  m = 1/N * (a1 + a2 + a3 + … +aN)

  g = (a1 * a2 * a3 * … * aN) ^ (1/N)
- Beispiel:

  Messreihe-A = 5, 5, 5, 5, 5, 5, 5, 5, 5, 5

  Messreihe-B = 1, 1, 1, 1, 41, 1, 1, 1, 1, 1

  SpectrumFlatness(A) = g(A) / m(A) = 5 / 5 = 1

  SpectrumFlatness(B) = g(B) / m(B) = (41^0.1) / 5 = 0.28

QUELLE: http://www.imk.fraunhofer.de/sixcms/media.php/208/hellmuth_audioid.pdf

- Beispiel-2: Fraunhofer AudioID: Erkennungsleistung
  - Datenbank: 15.000 Musikstücke
  - Extraktion aus dem Spektrum von 250 HZ – 1000Hz
  - Test: 1.000 Musikstücke; Länge 4s;
    - MP3@ 96kbps 99,8% korrekte Fingerprints
    - Mikrofon 97, 2%
    - Resampling 96,4%
    - Equalizer 99,3%
    - Dyn. Kompression 99,8%

QUELLE: http://www.imk.fraunhofer.de/sixcms/media.php/208/hellmuth_audioid.pdf

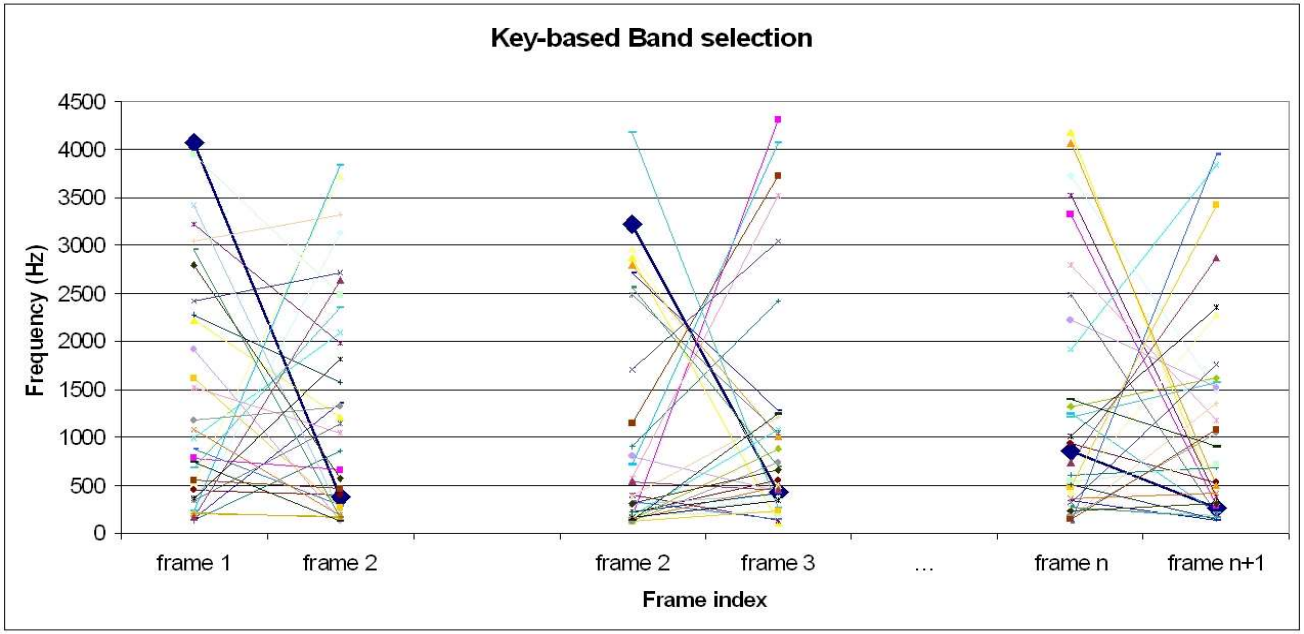
Passive Fingerabdrücke

- Weitere Anbieter von Fingerprinting Diensten:
  - Audible Magic
  - Musictrace
  - Phillips

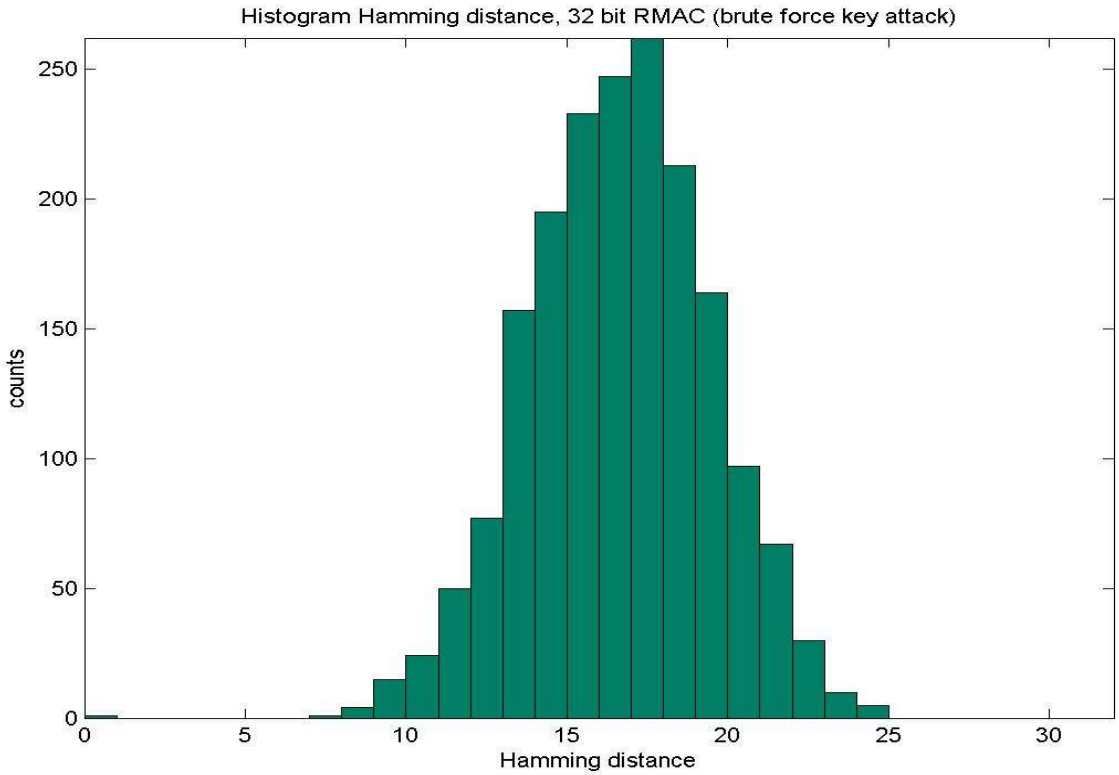
- Wie „sicher“ sind passive Fingerabdrücke
  - Kann eine Datei so verändert werden, dass sie noch gleich klingt, aber einen stark unterschiedlichen Fingerabdruck hat?
  - Kann man eine Datei so verändern, dass ihr Fingerabdruck der einer anderen Datei entspricht?

- Diese Fragestellungen sind derzeit noch als Forschungsgegenstand zu betrachten

- Kann man eine Datei so verändern, dass ihr Fingerabdruck der einer anderen Datei entspricht?
  - Lösungsansatz:
    - Permutation der zu vergleichenden Frequenzen mittels geheimen Schlüssel
    - Beispiel:
      - 64 Halbtonschritte von 110 Hz – 4435 Hz
      - 32 Bit robuster MAC



- Lösungsansatz:
  - Sicherheit gegen Erraten des Schlüssels
  - Hamming-Distanz des Hashs bei zufälligen Schlüsseln



**Test: JPEG mit Paint laden,
wieder als JPEG speichern.**

- E:\Temp>md5 *.jpg
- 3B3FD01FD259BD9E215F76821C7FD4A1 SANY0178.JPG
- FA54E4027DDCE7F315F78DEFE964B0ED SANY0178_1.JPG
- A6D7610ECBA10813394849775BB051CD SANY0178_2.JPG
- 8ACB44D55693F3811D28A6AE398F9BB5 SANY0178_3.JPG

SSDeep und JPEG

```
SANY0178_1.JPG: 2PnL4ySsdX4Y37h8N9YzSWTaEvq2zmhpf1JhModDmEQbA8BUl:
                65SSIEhhah2zef1JpdmdPBUl

SANY0178_2.JPG: 9qAjnknYytozR63RErlfjQbHGuIWjkqcDG:9qAjtytom2lfW6i
```

Robuste Hashverfahren

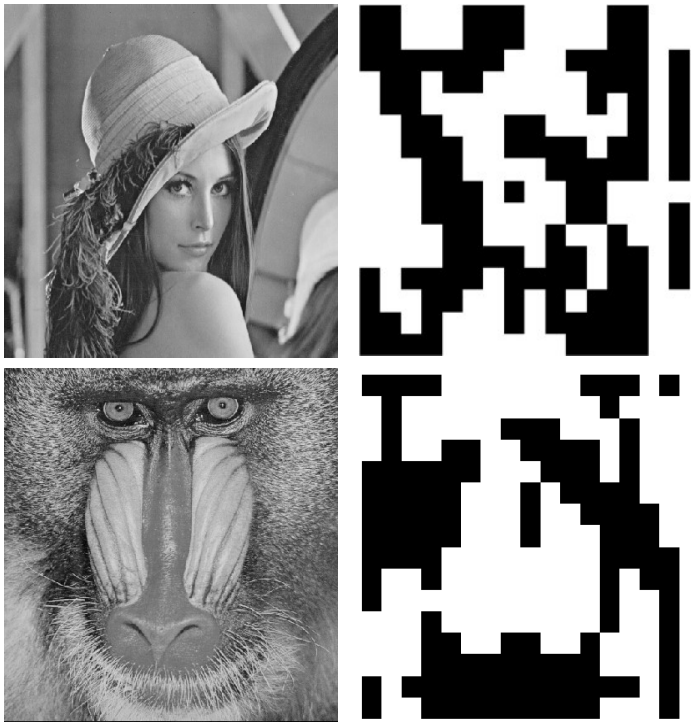
- Robuste Hashverfahren
  - sind aufwändiger als kryptographische Hashverfahren
    - Abhängig von Konzept und Implementierung kann schnell ein Zeitfaktor 1.000 erreicht werden
- sind weniger genau als kryptographische Hashverfahren
  - Fehlerraten (FAR/ FRR)
  - Robuste Hashs benötigen mehr Speicherplatz
- benötigen mehr Speicherplatz

Robuste Hashverfahren

- Beispiele für bekannte robuste Hashverfahren
  - DCT (Spektrum des Bildes)
  - Marr-Hildreth Operator (Kantenerkennung)
  - Radiale Transformation (Projektion)
  - Blockdurschnitt (Helligkeit von Blöcken)

Blockhash

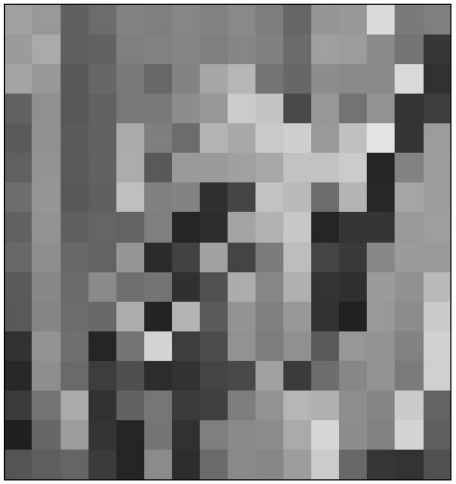
- Jeweils Bild und zugehöriger robuster Hash



Blockhash

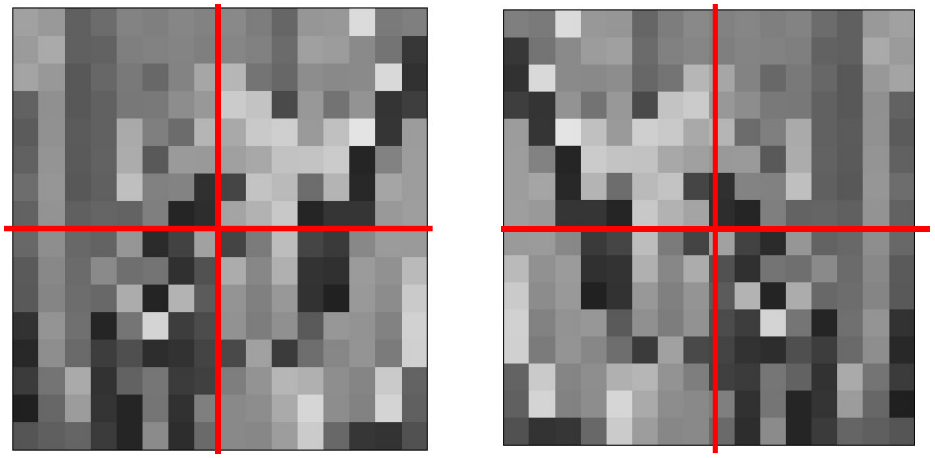
- Schritt 1:
  - Graustufen
  - Skalieren auf 16 x 16 Pixel





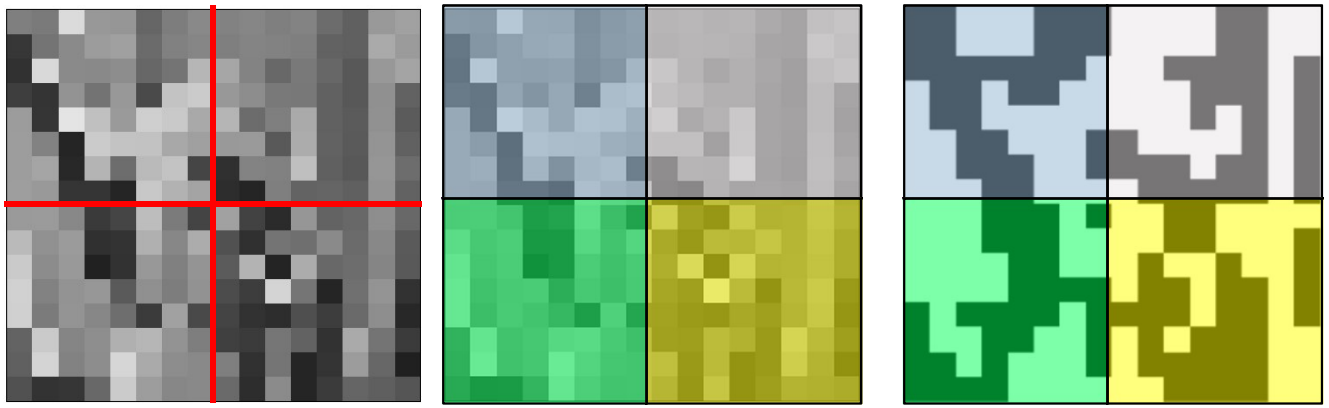
Blockhash

- Schritt 2:
  - Automatische Spiegelung
  - Quadrant mit hellsten Punkten oben links



Blockhash

- Schritt 3:
  - Pro Quadrant von 8x8 Pixeln Berechnung des Medians
  - Hashentscheidung: Pixel < oder >= Median



Blockhash

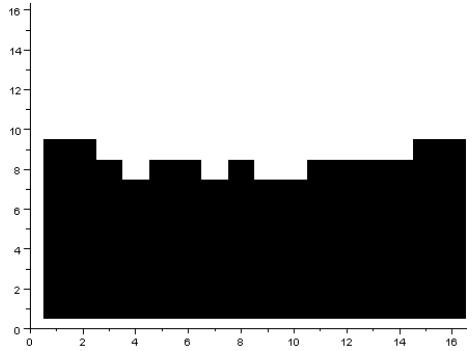
- Hashgröße
  - 16 x 16 Pixel, also 256 Bit
  - 2^256= 1,158*10^77
  - Geschätzte Anzahl von Atomen im Universum: 10^78
  - Bilder, die zufällig den geleichen Hash haben, sind also unwahrscheinlich
  - Wahrscheinlichkeit ist aber höher als oben berechnet, da Bilder wiederkehrende Strukturen haben, die sich in ähnlichen Hashsequenzen niederschlagen
    - Portraits
    - Sonnenaufgänge
    - Strand und Meer
    - …

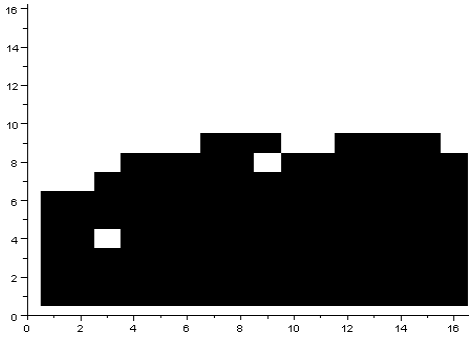
Blockhash

- Beispiel Struktur Problem
  - A: Fotos
  - B: Hash nach Rotation
    - Sehr ähnliche Struktur
  - C: Hash nach Aufteilen in 8x8 Blöcke
    - Abstand deutlich höher

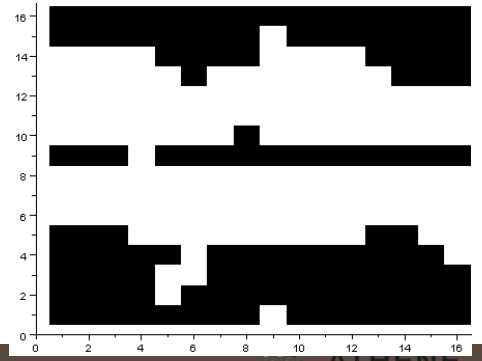
A

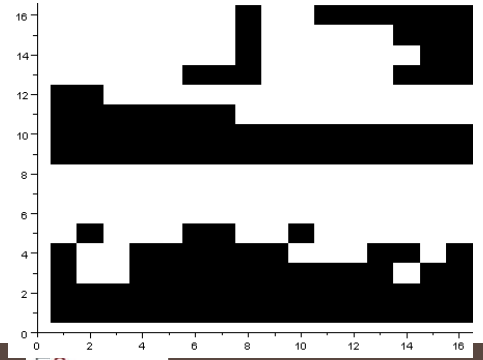
B





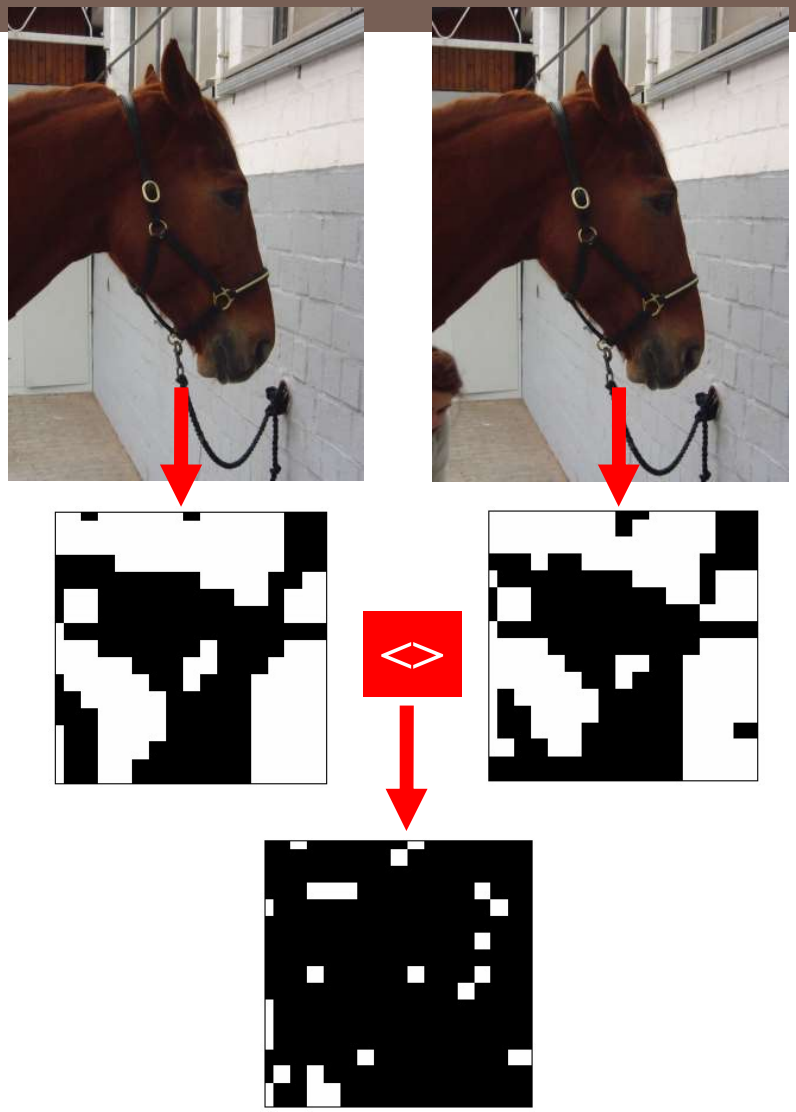
C





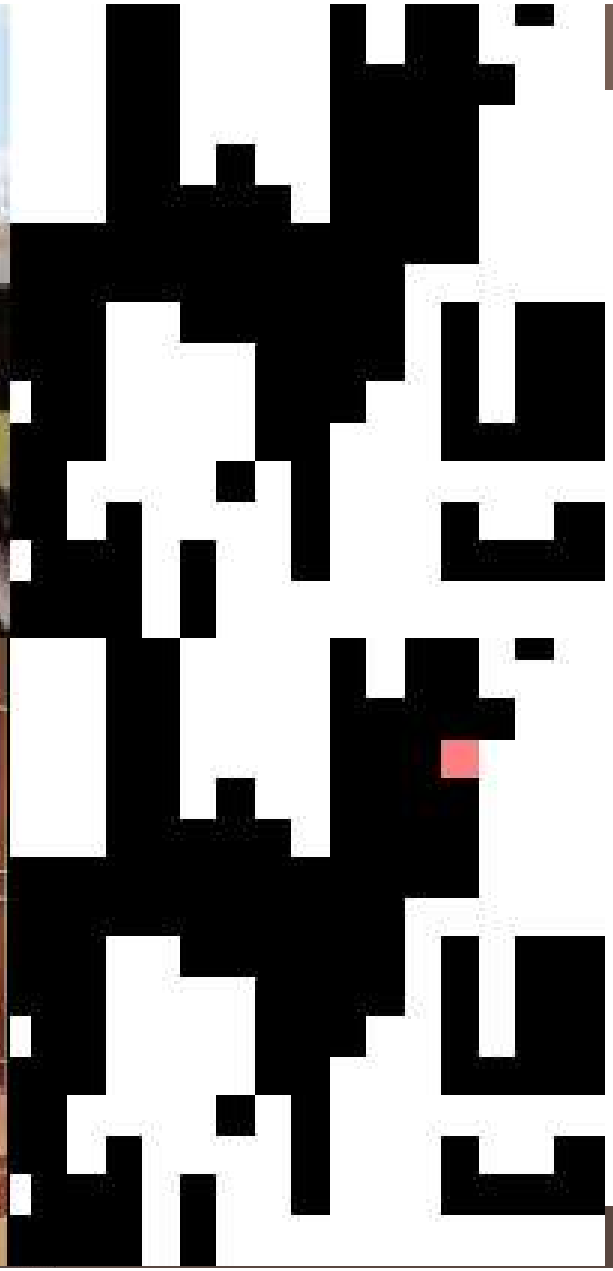
Blockhash

- Auch sehr ähnliche Bilder erzeugen deutlich unterschiedliche Hashes
  - Natürlich nicht so sehr wie Kryptographische Hashes
  - Ähnlichkeit von Bildern schlägt sich in ähnliche Hashes nieder



Blockhash

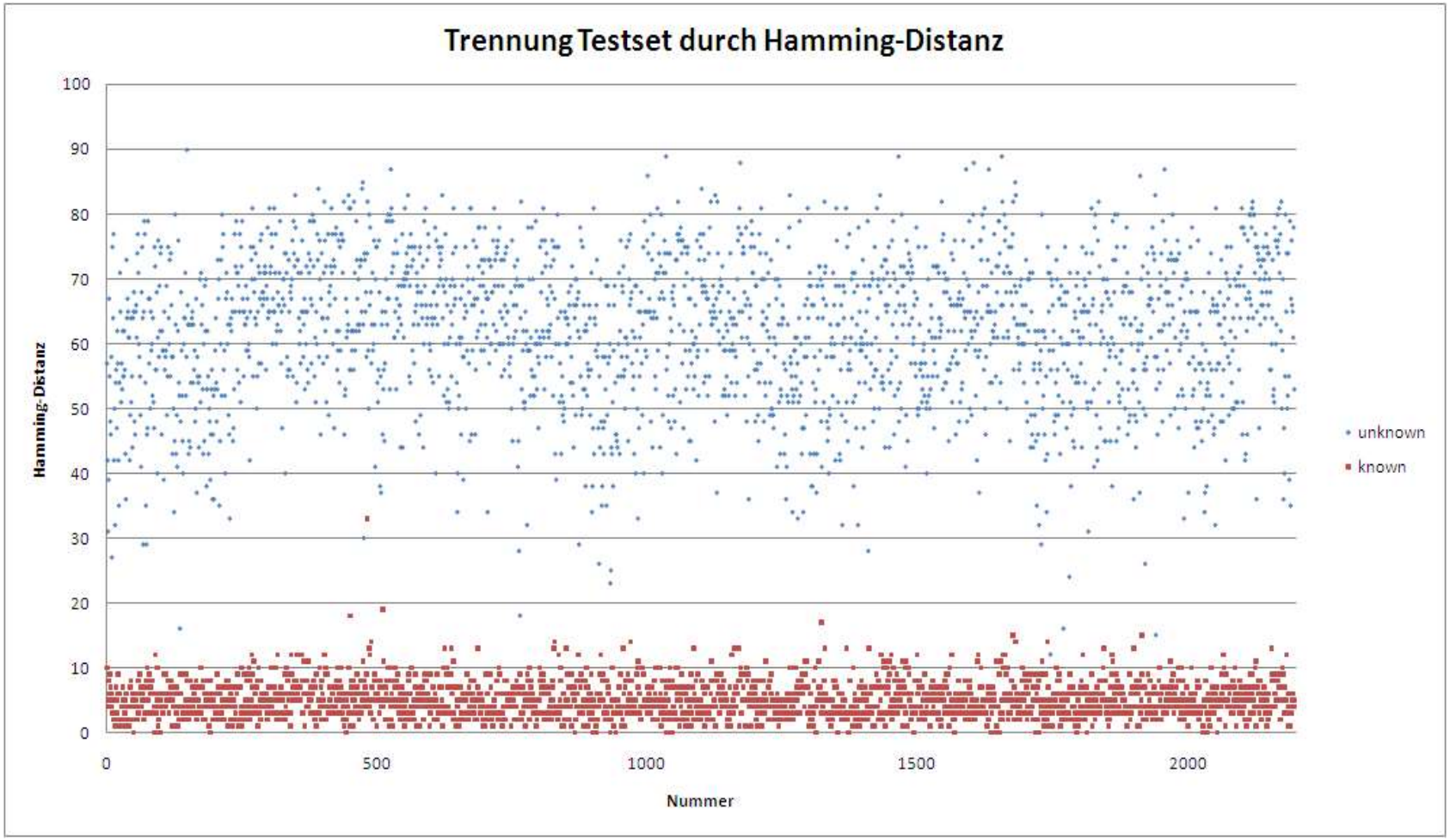
- Beispiel Robustheit
  - Starke JPEG Kompression
  - Spiegelung
  - Nur ein Bit Differenz



www.dollmanege.ru

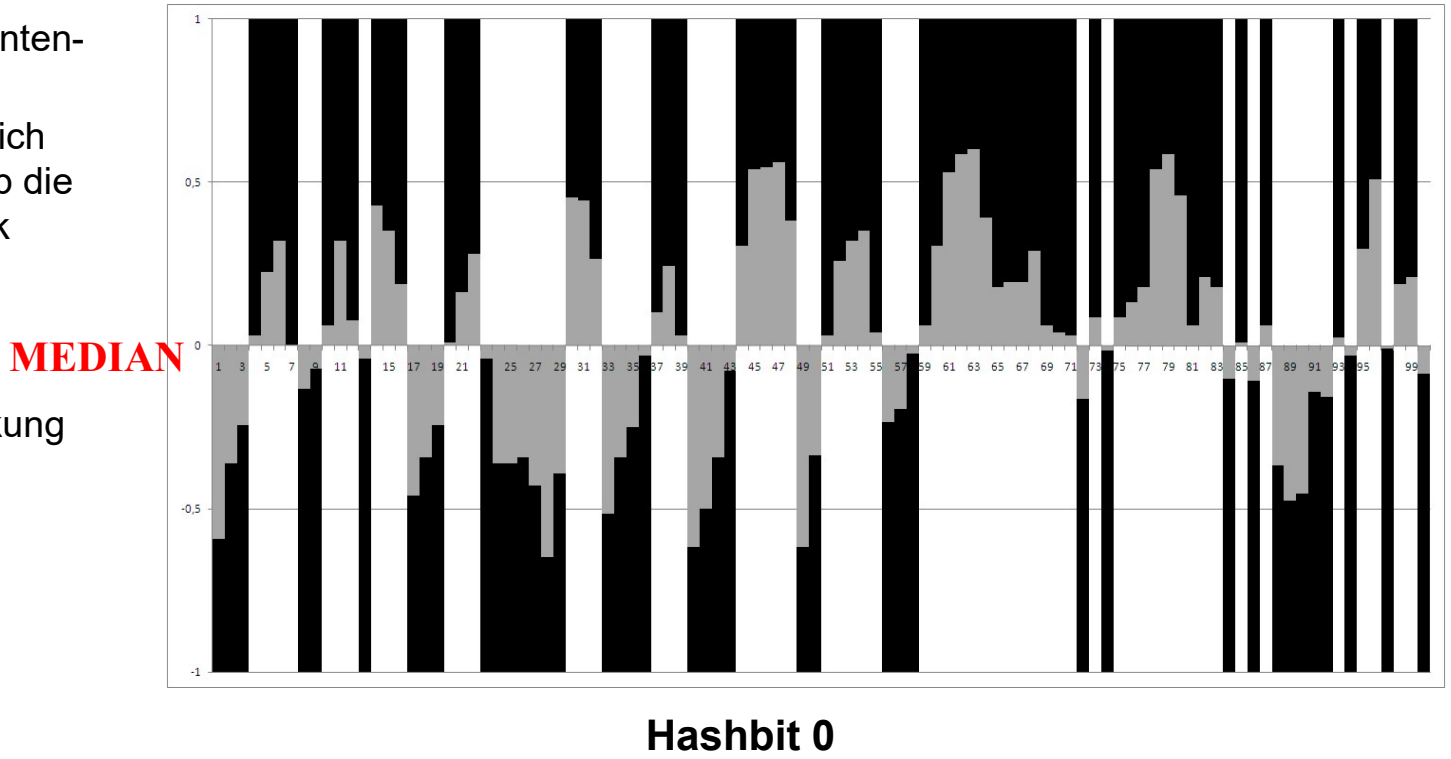
Blockhash

- Entscheidung, ob ein Hash zu einem bekannten Bild in der Datenbank gehört, wird über Hamming-Distanz gefällt
  - Anzahl der Bits, die sich zwischen Bildhash und Hash in Datenbank unterscheiden
- Im Gegensatz zu herkömmlichen kryptographischen Hashs wird nach ähnlichen Hashs gesucht, nicht nach identischen
- Beispiel:
  - Hashgröße 256 Bit
  - Bild wird erkannt, wenn es eine Hammingdistanz <=32 zu einem Bild in der Datenbank hat


Trennung Testset durch Hamming-Distanz
Hamming-Distanz
100
90
80
70
60
50
40
30
20
10
0
0
500
1000
1500
2000
Nummer
unknown
known

Blockhash

- Nicht jedes Hashbit ist gleich
  - Nicht gleich robust gegen Änderungen
  - Nicht gleich vertrauenswürdig
- Daraus resultiert das Konzept des Quanten-Hashes
  - Die Hamming-Distanz wird zusätzlich bewertet, indem betrachtet wird, ob die nicht übereinstimmenden Bits stark ausgeprägt waren oder nicht
    - Stark ausgeprägt: Unterschiedliches Bild
    - Schwach ausgeprägt: Auswirkung von Kompression o.ä.

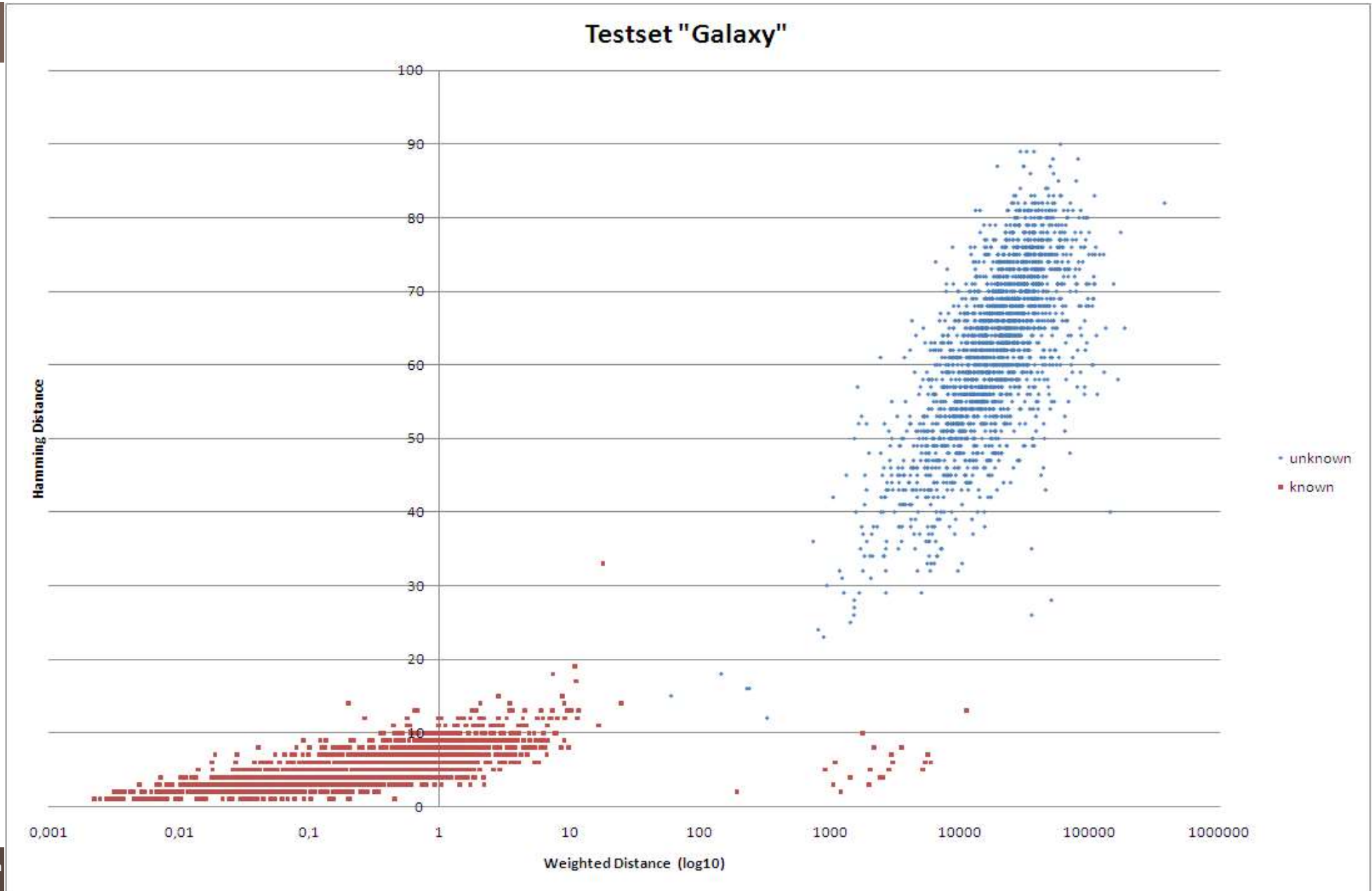


Blockhash

- Berechnung des Quantenhashes „gewichtete Hammingdistanz“

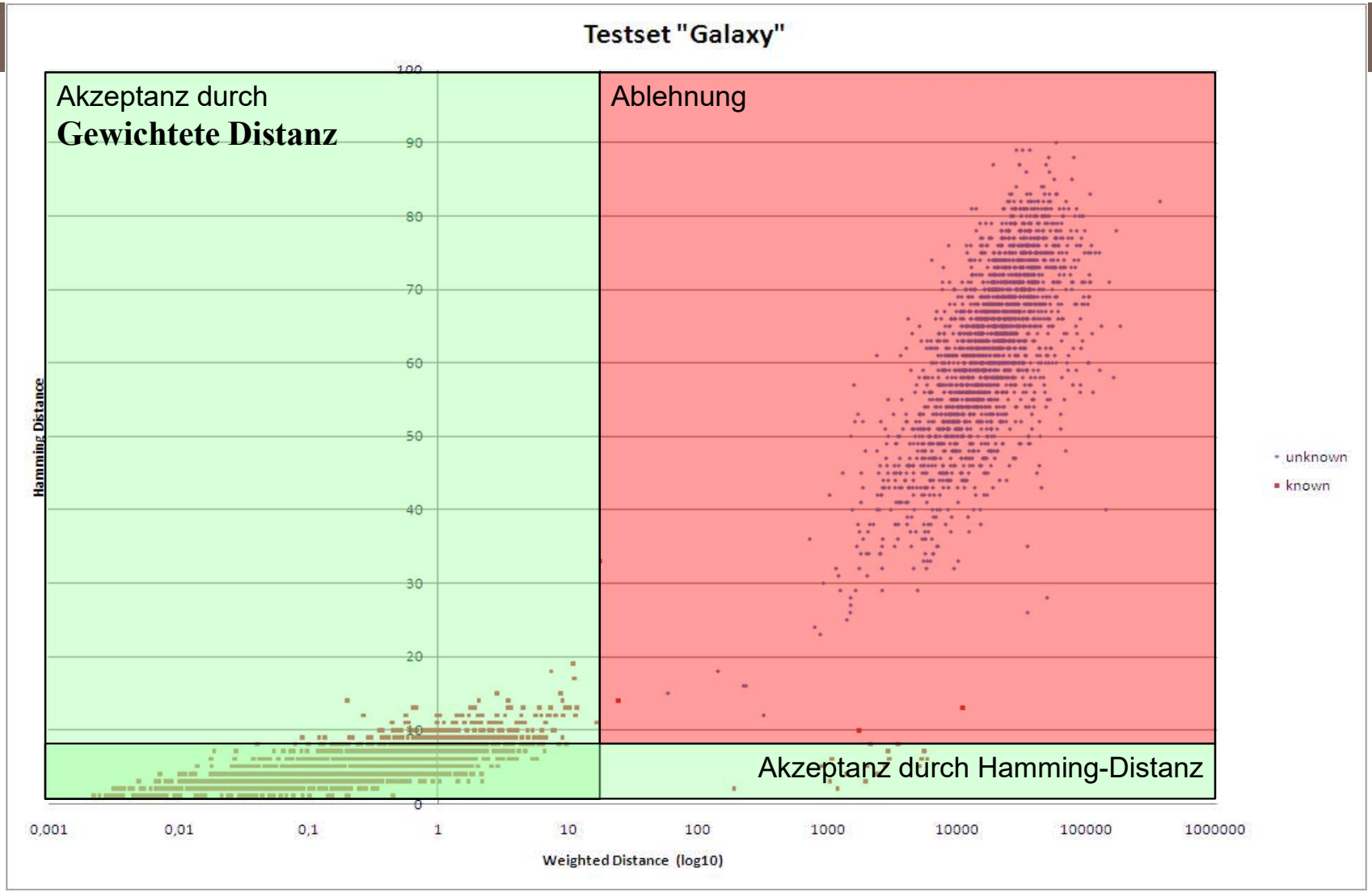
$$\frac{\text{Varianz(Abstände unterschiedlicher Hashbits zum Median der Blockhelligkeit)}}{\text{Varianz(Abstände gleicher Hashbits zum Median der Blockhelligkeit)}} * \text{Hammingdistanz} * 1000$$

- Beispiel folgende Seite:
  - X-Achse = Quantenhash
  - Y-Achse= Hammingdistanz


Testset "Galaxy"
Hamming Distance
100
90
80
70
60
50
40
30
20
10
0
0,001
0,01
0,1
1
10
100
1000
10000
100000
1000000
Weighted Distance (log10)
unknown
known

Blockhash

- Sehr gute FRR/FAR bei Kombination aus Hamming-Distanz und gewichteter Distanz
  - Alle Bilder mit Hamming-Distanz <= 32 werden betrachtet
  - Bilder mit Hamming-Distanz <=8 oder gewichteter Distanz <=16 werden akzeptiert
- Testszenario
  - Testset mit 4.394 Bildern
  - Hashbibliothek mit > 80.000 Bildern
  - Hälfte der Bilder in Bibliothek aufgenommen (known)
  - Andere Hälfte nicht aufgenommen (unknown)
- Ergebnisse
  - FAR = 0, FRR = 5
  - Fehlerrate also 1,1 Promille


Testset "Galaxy"
Akzeptanz durch
Gewichtete Distanz
Ablehnung
Akzeptanz durch Hamming-Distanz
Hamming Distance
Weighted Distance (log10)
unknown
known
0
10
20
30
40
50
60
70
80
90
100
0,001
0,01
0,1
1
10
100
1000
10000
100000
1000000

Testmengen (Beispiel)

Auch sehr ähnliche Bilder werden erfolgreich zugeordnet

© http://www.cheerleader-frankfurt.de/galacticdancers/

Blockhash Beschleunigen

- 40% der Zeit der Hashberechnung entfällt auf JPEG Dekodierung
- Hash skaliert von voller Auflösung auf 16x16
  - Rechenzeit steigt mit Bildgröße
- Oft ist das Verhältnis von Blockgröße und Bildauflösung extrem
  - 3200x1600 Auflösung wird zu 16x16
  - 200x100 Pixel pro Block
- Vereinfachung
  - Nur DC Koeffizient extrahieren
  - 8x8 Raster von Bild entsteht