

# Perceptual Hashing

Identifikation von Bildern aufgrund ihrer “Fingerabdrücke”

Dominik Schreiber

TU Darmstadt

Seminar Digitale Forensik, Sommersemester 2011, 16.06.2011

# Übersicht

## Bild-Forensik

Quellenidentifikation

Authentizitätsprüfung

Manipulationserkennung

Perceptual Hashing

# Übersicht

## Bild-Forensik

### Quellenidentifikation

Authentizitätsprüfung

Manipulationserkennung

Perceptual Hashing

# Übersicht

Bild-Forensik

Quellenidentifikation

Authentizitätsprüfung

Manipulationserkennung

Perceptual Hashing

# Warum das Ganze?



**Abbildung:** Iranisches Waffenprogramm: was nicht passt wird *passend gemacht*. Quelle: *Multimedia-Forensik als Teilgebiet der Digitalen Forensik*, Böhme et al., 28.09.2009

# Übersicht

## Bild-Forensik

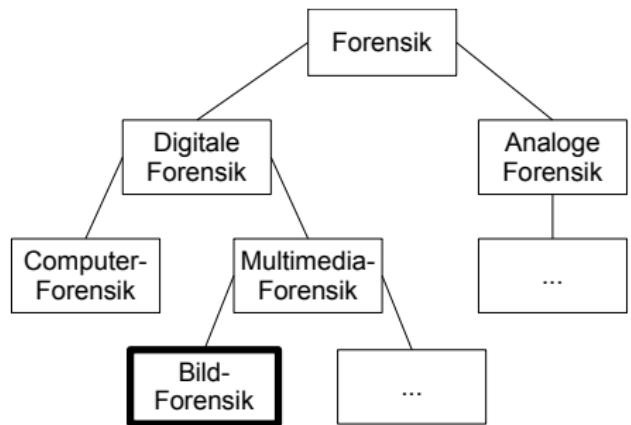
Quellenidentifikation

Authentizitätsprüfung

Manipulationserkennung

Perceptual Hashing

# Bild-Forensik innerhalb der digitalen Forensik



## Digitale Bildforensik

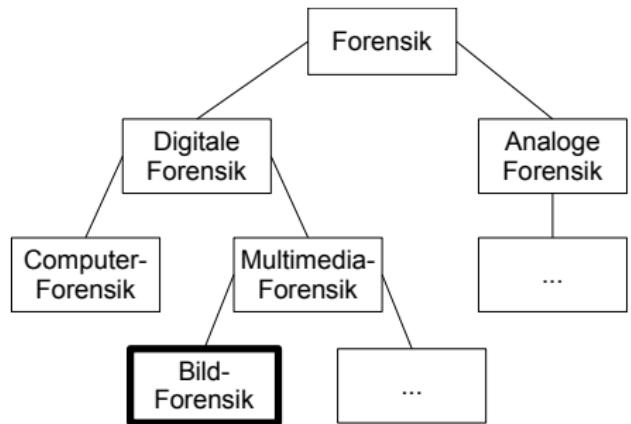
“Analyse von digitalen Bildern mit Hilfe von Computerprogrammen zur

- ▶ *Bestimmung des Bildursprungs oder der*
- ▶ *Aufdeckung von Bildmanipulationen.”<sup>a</sup>*

Abbildung: Strukturierung der forensischen Disziplinen

<sup>a</sup>Digitale Bildforensik,  
Gloe & Kirchner,  
TU Dresden, 18.6.2010

# Bild-Forensik innerhalb der digitalen Forensik



## Digitale Bildforensik

“Analyse von digitalen Bildern mit Hilfe von Computerprogrammen zur

- ▶ *Bestimmung des Bildursprungs oder der*
- ▶ *Aufdeckung von Bildmanipulationen.”<sup>a</sup>*

Abbildung: Strukturierung der forensischen Disziplinen

---

<sup>a</sup>*Digitale Bildforensik,*  
Gloe & Kirchner,  
TU Dresden, 18.6.2010

# Multimedia-Forensik vs. Computer-Forensik

## Multimedia-Forensik

- ▶ *Sensordaten* (Bild, Video, Audio)
- ▶ “perfektes Verbrechen” nahezu *unmöglich* (Sensordaten bilden Realität ab)

## Computer-Forensik

- ▶ *digitale Daten* (Logfiles, Netzwerk-Captures)
- ▶ “perfektes Verbrechen” theoretisch *möglich* (endlich viele Zustände)

# Übersicht

Bild-Forensik

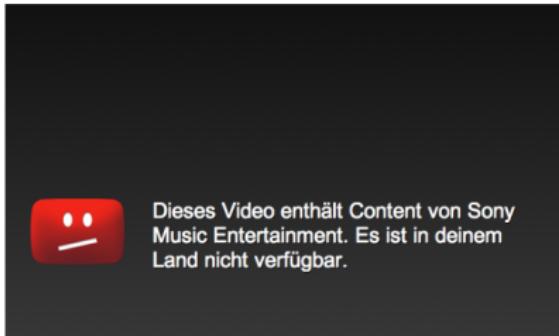
Quellenidentifikation

Authentizitätsprüfung

Manipulationserkennung

Perceptual Hashing

# Problem: Wer wars?



## Problembeschreibung

Bestimme bei einem digitalen Bild

- ▶ den *Hersteller* der Kamera,
- ▶ das *Modell* der Kamera,
- ▶ idealerweise genau *die eine* Kamera,

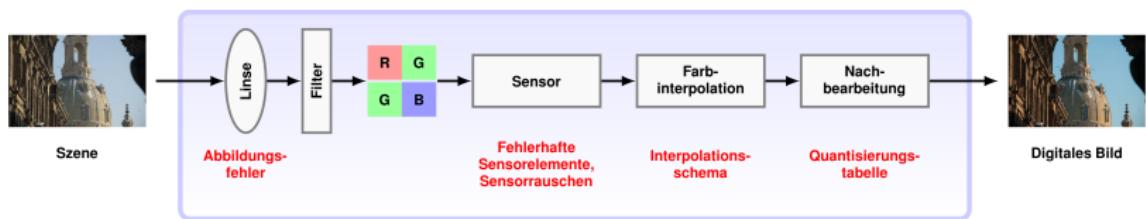
die es erstellt hat.

Abbildung: Finde den Urheber

# Lösungsmöglichkeiten I

## Grundsätzliche Idee

Verwende *Charakteristika* von Kameras/Scannern um Ursprung eines Bildes zu erkennen.



**Abbildung:** Mögliche Charakteristika bei einer Digitalkamera, Quelle:  
[http://commons.wikimedia.org/wiki/File:Digitalkamera\\_Bildforensik.png](http://commons.wikimedia.org/wiki/File:Digitalkamera_Bildforensik.png),  
Matthias Kirchner, 3.8.2009

# Lösungsmöglichkeiten II



**Abbildung:** Sensorrauschen,  
Quelle: [www.openfoto.de/tag/sensorrauschen](http://www.openfoto.de/tag/sensorrauschen)

## Gerätemodell

- ▶ Sensorrauschen
- ▶ Farbfilter, Farbinterpolationsalgorithmus
- ▶ interne Bildverarbeitungsschritte (z.B. Weißabgleich)

## Das einzelne Gerät

- ▶ *CCD/CMOS-Sensorrauschen*
- ▶ defekte Sensorelemente, Staubpartikelablagerungen

# Lösungsmöglichkeiten II



**Abbildung:** Sensorrauschen,  
Quelle: [www.openfoto.de/tag/sensorrauschen](http://www.openfoto.de/tag/sensorrauschen)

## Gerätemodell

- ▶ Sensorrauschen
- ▶ Farbfilter, Farbinterpolationsalgorithmus
- ▶ interne Bildverarbeitungsschritte (z.B. Weißabgleich)

## Das einzelne Gerät

- ▶ *CCD/CMOS-Sensorrauschen*
- ▶ defekte Sensorelemente, Staubpartikelablagerungen

# Übersicht

Bild-Forensik

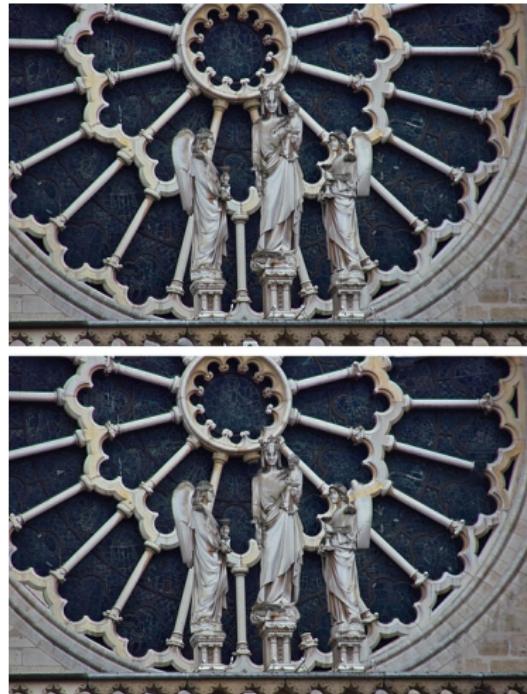
Quellenidentifikation

**Authentizitätsprüfung**

Manipulationserkennung

Perceptual Hashing

# Problem: Ist das Bild authentisch?



## Problembeschreibung

Weise bei einem digitalen Bild nach

- ▶ dass es *nicht sinnverändernd* nachbearbeitet wurde, oder
- ▶ *an welchen Stellen* es verändert wurde.

**Abbildung:** Finde die 10 Fehler

Quelle: [www.fotocommunity.de/pc/pc/display/23584881](http://www.fotocommunity.de/pc/pc/display/23584881)

# Manipulationserkennung

## Möglichkeiten zur Manipulationserkennung

- ▶ *lineare Abhängigkeiten* nach Resampling (wg. Interpolation)
- ▶ *zerstörte Pixel-Abhängigkeiten*, die es wg. CCD/CMOS-Sensoren geben müsste
- ▶ *charakteristische Periodizitäten* nach diskreter Kosinustransformation (DCT) bei JPEG-Kompression
- ▶ *zerstörtes Sensorrauschen* (siehe Folie 10)
- ▶ *sehr ähnliche Bildbereiche*, die größer als 0.85% der Bildgröße sind (Duplikate)

# Wann ist eine Veränderung auch *sinnverändernd*?

## Sinnverändernd (Manipulation):

kann so ziemlich alles sein, daher Beispiele:



Abbildung: (geringfügige) Manipulationen

## nicht sinnverändernd (Modifikation):

- ▶ Rotation
- ▶ Konvertierung
- ▶ Farbe
- ▶ Beschnitt
- ▶ Überlagerung
- ▶ Spiegelung
- ▶ Rauschen
- ▶ Filterung
- ▶ Komprimierung
- ▶ Verzerrung
- ▶ Kontrast

# Wann ist eine Veränderung auch *sinnverändernd*?

## Sinnverändernd (Manipulation):

kann so ziemlich alles sein, daher Beispiele:

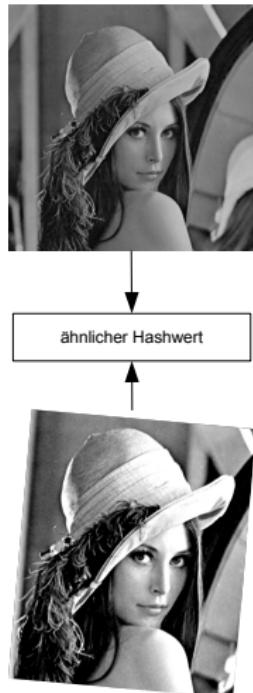


Abbildung: (geringfügige) Manipulationen

## nicht sinnverändernd (Modifikation):

- ▶ Rotation
- ▶ Konvertierung
- ▶ Farbe
- ▶ Beschnitt
- ▶ Überlagerung
- ▶ Spiegelung
- ▶ Rauschen
- ▶ Filterung
- ▶ Komprimierung
- ▶ Verzerrung
- ▶ Kontrast

# Authentizitätsprüfung



## Idee

- ▶ weise die Authentizität von Bildern (nur modifiziert, nicht manipuliert) mit *Hashwerten* nach
- ▶ vom menschlichen Auge als ähnlich erkannte Bilder sollen auf *denselben* (oder einen hinreichend ähnlichen<sup>a)</sup>) *Hashwert* abgebildet werden

---

<sup>a</sup>bzgl. einer gewählten *Metrik* und einem gewählten *Schwellenwert*

**Abbildung:** ähnliche Bilder →  
ähnlicher Hashwert

# Perceptual Hashing

## Problem: *Kryptographisches* Hashing

- ▶ Kryptographische Hashfunktionen (md5, sha-1) hängen signifikant von *jedem Bit* der Eingabe ab
- ▶ schon 1 Pixel Unterschied ruft einen *völlig unterschiedlichen* Hashwert hervor

## Lösung: *Wahrnehmungsba siertes* Hashing<sup>1</sup>

- ▶ erstelle Vektor mit allen relevanten Merkmalen des Bildes (→ Hashwert)
- ▶ vergleiche Vektoren mit vorher gewählter Metrik (z.B. Euklidische Distanz, Manhattan Distance)
- ▶ liegt das Ergebnis unterhalb eines vorher gewählten Schwellenwerts werden verglichene Bilder als gleich angesehen

---

<sup>1</sup> auch Perceptual Hashing, Robust Hashing

# Perceptual Hashing

## Problem: *Kryptographisches* Hashing

- ▶ Kryptographische Hashfunktionen (md5, sha-1) hängen signifikant von *jedem Bit* der Eingabe ab
- ▶ schon 1 Pixel Unterschied ruft einen *völlig unterschiedlichen* Hashwert hervor

## Lösung: *Wahrnehmungsba siertes* Hashing<sup>1</sup>

- ▶ erstelle *Vektor mit allen relevanten Merkmalen* des Bildes (→ Hashwert)
- ▶ vergleiche Vektoren mit vorher gewählter *Metrik* (z.B. Euklidische Distanz, Manhattan Distance)
- ▶ liegt das Ergebnis unterhalb eines vorher gewählten *Schwellenwerts* werden verglichene Bilder als *gleich* angesehen

---

<sup>1</sup>auch *Perceptual Hashing, Robust Hashing*

# Anforderungen an einen Perceptual-Hashing-Algorithmus<sup>2</sup> I

Sei

- ▶  $\mathcal{I}$  eine endliche Menge von Bildern
- ▶  $\mathcal{K}$  die Menge der privaten Schlüssel
- ▶  $H : \mathcal{I} \times \mathcal{K} \mapsto \{0, 1\}^q$  die Hashfunktion, die einen Hashwert der Länge  $q$  liefert
- ▶ zu  $I \in \mathcal{I}$   $I_{ident} \in \mathcal{I}$  ein Bild, das genau so aussieht
- ▶ zu  $I \in \mathcal{I}$   $I_{diff} \in \mathcal{I}$  ein Bild, das anders aussieht
- ▶  $0 < \theta_1, \theta_2 < 1$  fixe Werte, möglichst klein

---

<sup>2</sup>Quelle: *A Clustering Based Approach to Perceptual Image Hashing*, Monga et al., März 2006

# Anforderungen an einen Perceptual-Hashing-Algorithmus<sup>2</sup> I

Sei

- ▶  $\mathcal{I}$  eine endliche Menge von Bildern
- ▶  $\mathcal{K}$  die Menge der privaten Schlüssel
- ▶  $H : \mathcal{I} \times \mathcal{K} \mapsto \{0, 1\}^q$  die Hashfunktion, die einen Hashwert der Länge  $q$  liefert
- ▶ zu  $I \in \mathcal{I}$   $I_{ident} \in \mathcal{I}$  ein Bild, das genau so aussieht
- ▶ zu  $I \in \mathcal{I}$   $I_{diff} \in \mathcal{I}$  ein Bild, das anders aussieht
- ▶  $0 < \theta_1, \theta_2 < 1$  fixe Werte, möglichst klein

---

<sup>2</sup>Quelle: *A Clustering Based Approach to Perceptual Image Hashing*, Monga et al., März 2006

# Anforderungen an einen Perceptual-Hashing-Algorithmus<sup>2</sup> I

Sei

- ▶  $\mathcal{I}$  eine endliche Menge von Bildern
- ▶  $\mathcal{K}$  die Menge der privaten Schlüssel
- ▶  $H : \mathcal{I} \times \mathcal{K} \mapsto \{0, 1\}^q$  die Hashfunktion, die einen Hashwert der Länge  $q$  liefert
- ▶ zu  $I \in \mathcal{I}$   $I_{ident} \in \mathcal{I}$  ein Bild, das genau so aussieht
- ▶ zu  $I \in \mathcal{I}$   $I_{diff} \in \mathcal{I}$  ein Bild, das anders aussieht
- ▶  $0 < \theta_1, \theta_2 < 1$  fixe Werte, möglichst klein

---

<sup>2</sup>Quelle: *A Clustering Based Approach to Perceptual Image Hashing*, Monga et al., März 2006

# Anforderungen an einen Perceptual-Hashing-Algorithmus<sup>2</sup> I

Sei

- ▶  $\mathcal{I}$  eine endliche Menge von Bildern
- ▶  $\mathcal{K}$  die Menge der privaten Schlüssel
- ▶  $H : \mathcal{I} \times \mathcal{K} \mapsto \{0, 1\}^q$  die Hashfunktion, die einen Hashwert der Länge  $q$  liefert
- ▶ zu  $I \in \mathcal{I}$   $I_{ident} \in \mathcal{I}$  ein Bild, das genau so aussieht
- ▶ zu  $I \in \mathcal{I}$   $I_{diff} \in \mathcal{I}$  ein Bild, das anders aussieht
- ▶  $0 < \theta_1, \theta_2 < 1$  fixe Werte, möglichst klein

---

<sup>2</sup>Quelle: *A Clustering Based Approach to Perceptual Image Hashing*, Monga et al., März 2006

# Anforderungen an einen Perceptual-Hashing-Algorithmus<sup>3</sup> II

## Robustheit (wenige False-Negatives)

$$p(H(I, K) = H(I_{\text{ident}}, K)) \geq 1 - \theta_1$$

bei identisch wahrgenommenen Bildern ist der Hashwert mit einer Wahrscheinlichkeit von wenigstens  $1 - \theta_1$  gleich

## Wenige False-Positives

$$p(H(I, K) \neq H(I_{\text{diff}}, K)) \geq 1 - \theta_2$$

bei unterschiedlich wahrgenommenen Bildern ist der Hashwert mit einer Wahrscheinlichkeit von wenigstens  $1 - \theta_2$  unterschiedlich

## Unvorhersagbarkeit des Hashs

$\forall v \in \{0, 1\}^q : p(H(I, K) = v) \approx \frac{1}{2^q}$  jeder Hashwert ist gleich wahrscheinlich

---

<sup>3</sup>Anforderungen an den “idealen Hashalgorithmus” – dessen Existenz ist fraglich

# Anforderungen an einen Perceptual-Hashing-Algorithmus<sup>3</sup> II

## Robustheit (wenige False-Negatives)

$$p(H(I, K) = H(I_{\text{ident}}, K)) \geq 1 - \theta_1$$

bei identisch wahrgenommenen Bildern ist der Hashwert mit einer Wahrscheinlichkeit von wenigstens  $1 - \theta_1$  gleich

## Wenige False-Positives

$$p(H(I, K) \neq H(I_{\text{diff}}, K)) \geq 1 - \theta_2$$

bei unterschiedlich wahrgenommenen Bildern ist der Hashwert mit einer Wahrscheinlichkeit von wenigstens  $1 - \theta_2$  unterschiedlich

## Unvorhersagbarkeit des Hashs

$\forall v \in \{0, 1\}^q : p(H(I, K) = v) \approx \frac{1}{2^q}$  jeder Hashwert ist gleich wahrscheinlich

---

<sup>3</sup>Anforderungen an den “idealen Hashalgorithmus” – dessen Existenz ist fraglich

# Anforderungen an einen Perceptual-Hashing-Algorithmus<sup>3</sup> II

## Robustheit (wenige False-Negatives)

$$p(H(I, K) = H(I_{\text{ident}}, K)) \geq 1 - \theta_1$$

bei identisch wahrgenommenen Bildern ist der Hashwert mit einer Wahrscheinlichkeit von wenigstens  $1 - \theta_1$  gleich

## Wenige False-Positives

$$p(H(I, K) \neq H(I_{\text{diff}}, K)) \geq 1 - \theta_2$$

bei unterschiedlich wahrgenommenen Bildern ist der Hashwert mit einer Wahrscheinlichkeit von wenigstens  $1 - \theta_2$  unterschiedlich

## Unvorhersagbarkeit des Hashs

$\forall v \in \{0, 1\}^q : p(H(I, K) = v) \approx \frac{1}{2^q}$  jeder Hashwert ist gleich wahrscheinlich

---

<sup>3</sup>Anforderungen an den “idealen Hashalgorithmus” – dessen Existenz ist fraglich

# Anforderungen an einen Perceptual-Hashing-Algorithmus<sup>3</sup> II

## Robustheit (wenige False-Negatives)

$$p(H(I, K) = H(I_{\text{ident}}, K)) \geq 1 - \theta_1$$

bei identisch wahrgenommenen Bildern ist der Hashwert mit einer Wahrscheinlichkeit von wenigstens  $1 - \theta_1$  gleich

## Wenige False-Positives

$$p(H(I, K) \neq H(I_{\text{diff}}, K)) \geq 1 - \theta_2$$

bei unterschiedlich wahrgenommenen Bildern ist der Hashwert mit einer Wahrscheinlichkeit von wenigstens  $1 - \theta_2$  unterschiedlich

## Unvorhersagbarkeit des Hashs

$\forall v \in \{0, 1\}^q : p(H(I, K) = v) \approx \frac{1}{2^q}$  jeder Hashwert ist gleich wahrscheinlich

---

<sup>3</sup>Anforderungen an den “idealen Hashalgorithmus” – dessen Existenz ist fraglich

# Klassen von Algorithmen

## Klassifikation der Algorithmen nach ihrem Ansatz

- ▶ Invarianzen von *Bild-Statistiken*
- ▶ Invarianzen von *Bild-Relationen*
- ▶ Beibehalten der *groben Bild-Repräsentation*
- ▶ *Low-level* Merkmals-Extraktion

## Häufige Bestandteile

- ▶ Skalierung auf feste Größe
- ▶ Konvertierung in *Graustufen*
- ▶ Einteilen des Bildes in Blöcke
- ▶ (Diskrete) Kosinus-/Wavelet-/Fouriertransformation

# Klassen von Algorithmen

## Klassifikation der Algorithmen nach ihrem Ansatz

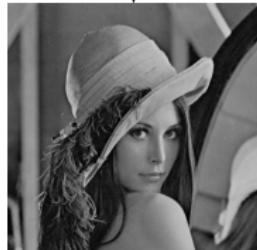
- ▶ Invarianzen von *Bild-Statistiken*
- ▶ Invarianzen von *Bild-Relationen*
- ▶ Beibehalten der *groben Bild-Repräsentation*
- ▶ *Low-level* Merkmals-Extraktion

## Häufige Bestandteile

- ▶ *Skalierung* auf feste Größe
- ▶ Konvertierung in *Graustufen*
- ▶ Einteilen des Bildes in Blöcke
- ▶ (Diskrete) Kosinus-/Wavelet-/Fouriertransformation

# Beispiel: Block Mean Value Based<sup>4</sup>

## Der Algorithmus



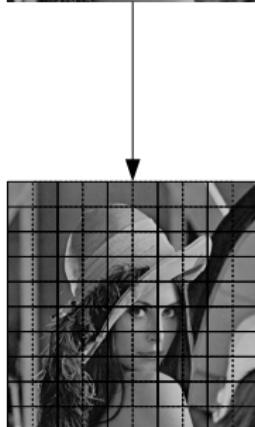
1. Bild auf vorgegebene Größe *normalisieren*
2. In  $N$  sich zur Hälfte überlappende Blöcke  $\{I_1, \dots, I_N\}$  aufteilen ( $N$  ist die Länge des Hashwerts)
3.  $\{I_1, \dots, I_N\}$  mit Schlüssel  $K$  verschlüsseln: neue Block-Sequenz  $\{I'_1, \dots, I'_N\}$
4. Arithmetische Mittelwerte  $\{M_1, \dots, M_N\}$  der Blöcke  $\{I'_1, \dots, I'_N\}$  berechnen, Median aller Mittelwerte berechnen:  
 $Md = median(\{M_1, \dots, M_N\})$
5. Hashwert  $h$  berechnen als  
$$h(i) = \begin{cases} 0 & M_i < Md \\ 1 & M_i \geq Md \end{cases}$$

---

<sup>4</sup>Quelle: *Block Mean Value Based Image Perceptual Hashing for Content Identification*, Yang et al., 2006

# Beispiel: Block Mean Value Based<sup>4</sup>

## Der Algorithmus



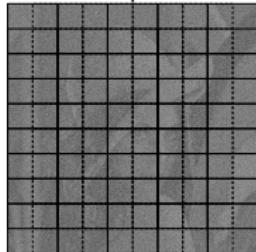
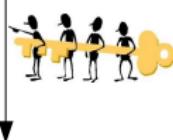
1. Bild auf vorgegebene Größe *normalisieren*
2. In  $N$  sich zur Hälfte überlappende Blöcke  $\{I_1, \dots, I_N\}$  aufteilen ( $N$  ist die Länge des Hashwerts)
3.  $\{I_1, \dots, I_N\}$  mit Schlüssel  $K$  verschlüsseln: neue Block-Sequenz  $\{I'_1, \dots, I'_N\}$
4. Arithmetische Mittelwerte  $\{M_1, \dots, M_N\}$  der Blöcke  $\{I'_1, \dots, I'_N\}$  berechnen, Median aller Mittelwerte berechnen:  
 $Md = median(\{M_1, \dots, M_N\})$
5. Hashwert  $h$  berechnen als  
$$h(i) = \begin{cases} 0 & M_i < Md \\ 1 & M_i \geq Md \end{cases}$$

---

<sup>4</sup>Quelle: *Block Mean Value Based Image Perceptual Hashing for Content Identification*, Yang et al., 2006

# Beispiel: Block Mean Value Based<sup>4</sup>

## Der Algorithmus

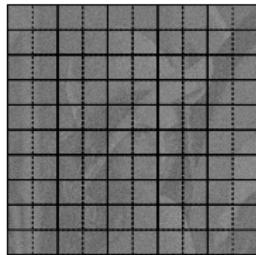


1. Bild auf vorgegebene Größe *normalisieren*
2. In  $N$  sich zur Hälfte überlappende Blöcke  $\{I_1, \dots, I_N\}$  aufteilen ( $N$  ist die Länge des Hashwerts)
3.  $\{I_1, \dots, I_N\}$  mit Schlüssel  $K$  verschlüsseln: neue Block-Sequenz  $\{I'_1, \dots, I'_N\}$
4. Arithmetische Mittelwerte  $\{M_1, \dots, M_N\}$  der Blöcke  $\{I'_1, \dots, I'_N\}$  berechnen, Median aller Mittelwerte berechnen:  
 $Md = median(\{M_1, \dots, M_N\})$
5. Hashwert  $h$  berechnen als  
$$h(i) = \begin{cases} 0 & M_i < Md \\ 1 & M_i \geq Md \end{cases}$$

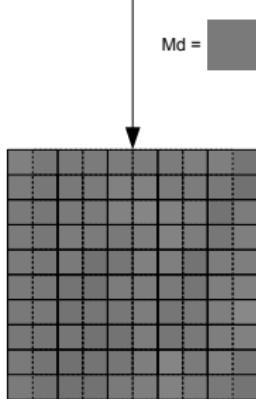
<sup>4</sup>Quelle: *Block Mean Value Based Image Perceptual Hashing for Content Identification*, Yang et al., 2006

# Beispiel: Block Mean Value Based<sup>4</sup>

## Der Algorithmus



Md =

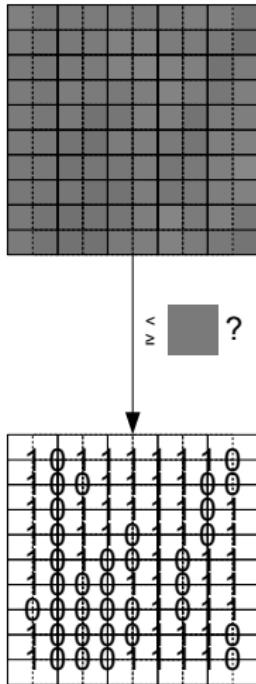


1. Bild auf vorgegebene Größe *normalisieren*
2. In  $N$  sich zur Hälfte überlappende Blöcke  $\{I_1, \dots, I_N\}$  aufteilen ( $N$  ist die Länge des Hashwerts)
3.  $\{I_1, \dots, I_N\}$  mit Schlüssel  $K$  verschlüsseln: neue Block-Sequenz  $\{I'_1, \dots, I'_N\}$
4. Arithmetische Mittelwerte  $\{M_1, \dots, M_N\}$  der Blöcke  $\{I'_1, \dots, I'_N\}$  berechnen, Median aller Mittelwerte berechnen:  
$$Md = median(\{M_1, \dots, M_N\})$$
5. Hashwert  $h$  berechnen als  
$$h(i) = \begin{cases} 0 & M_i < Md \\ 1 & M_i \geq Md \end{cases}$$

<sup>4</sup>Quelle: *Block Mean Value Based Image Perceptual Hashing for Content Identification*, Yang et al., 2006

# Beispiel: Block Mean Value Based<sup>4</sup>

## Der Algorithmus



1. Bild auf vorgegebene Größe *normalisieren*
2. In  $N$  sich zur Hälfte überlappende Blöcke  $\{I_1, \dots, I_N\}$  aufteilen ( $N$  ist die Länge des Hashwerts)
3.  $\{I_1, \dots, I_N\}$  mit Schlüssel  $K$  verschlüsseln: neue Block-Sequenz  $\{I'_1, \dots, I'_N\}$
4. Arithmetische Mittelwerte  $\{M_1, \dots, M_N\}$  der Blöcke  $\{I'_1, \dots, I'_N\}$  berechnen, Median aller Mittelwerte berechnen:  
 $Md = median(\{M_1, \dots, M_N\})$
5. Hashwert  $h$  berechnen als  
$$h(i) = \begin{cases} 0 & M_i < Md \\ 1 & M_i \geq Md \end{cases}$$

<sup>4</sup>Quelle: *Block Mean Value Based Image Perceptual Hashing for Content Identification*, Yang et al., 2006

# Weitere Anwendungsgebiete für Perceptual Hashing

## Bilddatenbanken

- ▶ Zu jedem vorhandenen Bild ist der *Hashwert gespeichert*
- ▶ Wird nun ein Bild eingefügt, kann der Hashwert gebildet werden und mit allen vorhandenen verglichen werden
- ▶ so lassen sich Duplikate effizient verhindern oder Bilder suchen

## Digitale Signaturen

Um Copyrights durchzusetzen können

- ▶ *Watermarks* in Bilder eingebettet werden, oder
- ▶ *Digitale Signaturen* mit Perceptual-Hashing-Algorithmen erstellt werden

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?