UNIVERSITÄT SALZBURG

Selektive Kompression von Iris Bildern und Matching Performance

Christian Prähauser

Thomas Starzacher

Inhalte

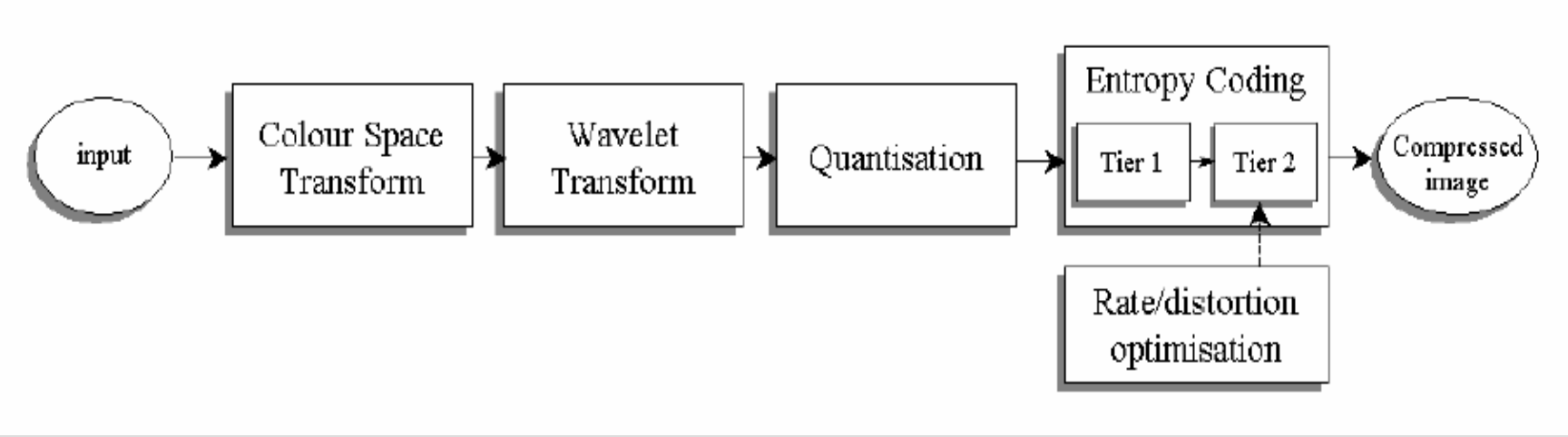
- Einführung in JPEG2000
- JPEG2000 Regions of Interest
- EyeDROID
  - Irisbestimmung
    - Canny Edge Detection
    - Hough Circle Detection
  - Enkodierung (JPEG2000)
- Iriserkennung
  - Matlab SW von Libor Masek
- Resultate
  - Matching Performance

JPEG2000 Allgemein

JPEG2000 (J2K)

- Entwickelt von der Joint Photographic Experts Group (JPEG)
- ISO/ITU-T Standard 15444-1:2000
- Basiert auf Diskreter Wavelet Transformation (DWT)
- Verlustfreie and verlustbehaftete Kompression
- Progressive
  - by resolution
  - by quality/pixel accuracy
  - **by Region ➔ Regions of Interest (ROI)**

JPEG2000 (J2K): Enkodierung



- Komponenten Transformation
- Wavelet Transformation
- Quantisierung
- Entropiekodierung
  - EBCOT

JPEG2000: Enkodierung

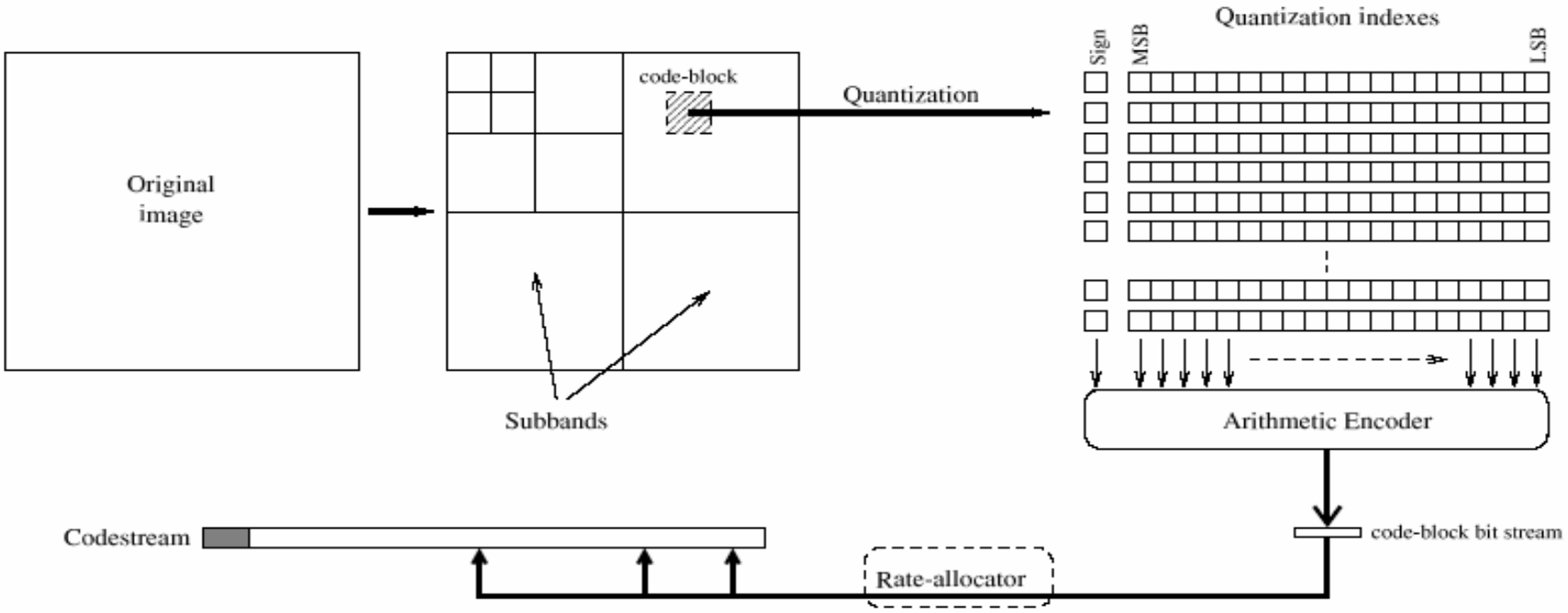
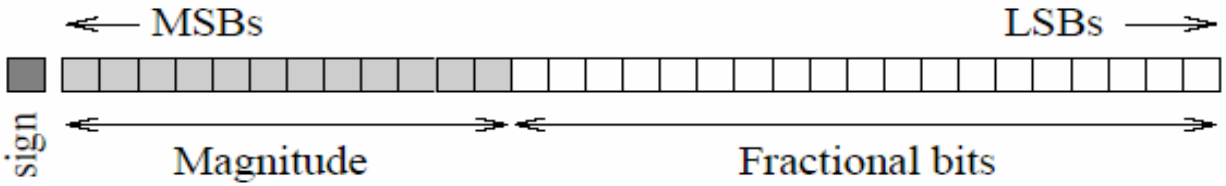


Figure 2. Example of sign-magnitude representation of a quantization index on 32 bits.



JPEG2000: Wavelet Transformation

- Mittels Wavelet Transformation (WT) wird das Bild in Subbands aufgeteilt
  - Jedes Subband enthält unterschiedliche Frequenzanteile
  - Dyadic decomposition
    - Wiederholtes Anwenden von low- und high-pass Filtern auf das Bild in beiden räumlichen Richtungen (Vertikal/Horizontal)
  - Abhängig von der Art des verwendeten Wavelets, kann verlustfrei (lossless) und verlustbehaftet (lossy) komprimiert werden
    - CDF 5/3 (Fixed-point) Wavelet ➔ lossless
    - CDF 9/7 (Floating-point) Wavelet ➔ lossy

JPEG2000: Entropiekodierung

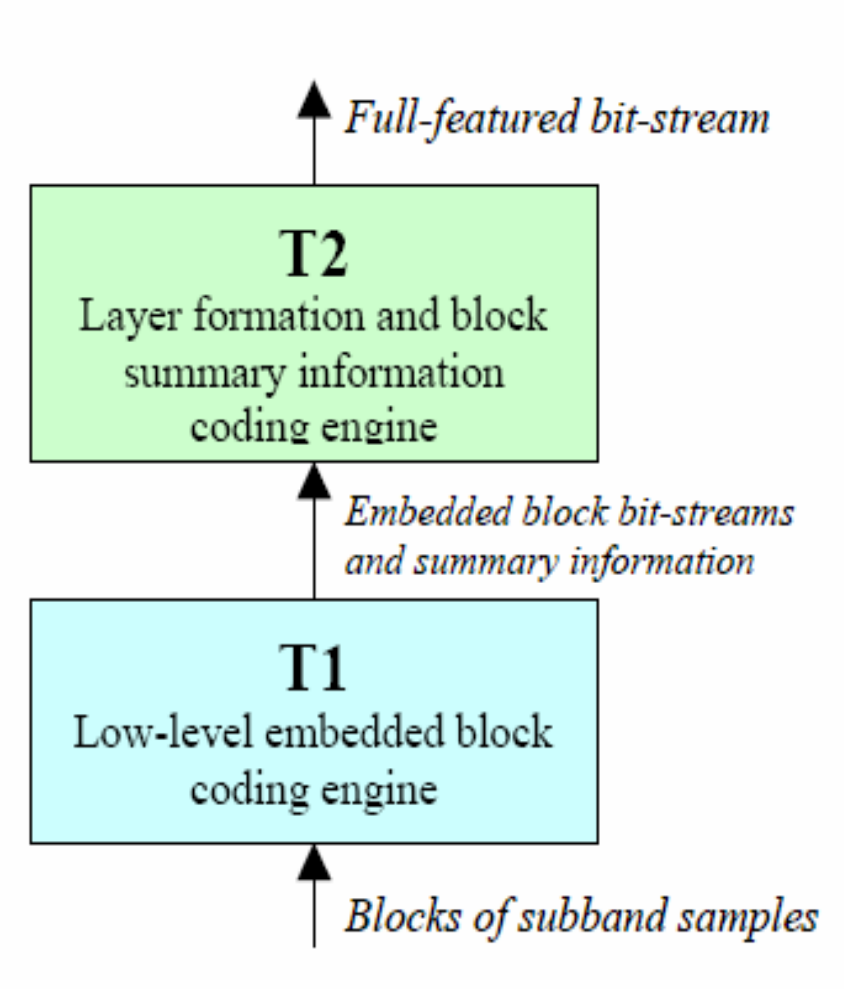
- Embedded Block Coding with Optimized Truncation (EBCOT)
- Die Koeff. (oder Indizes aus dem Quantisierer) werden (verlustfrei) entropiekodiert
  - Jedes Subband wird in Blöcke aufgeteilt (code blocks)
  - Die code blocks werden unabhängig voneinander kodiert
  - Die Kodierung beginnt bei der höchsten (most-significant) Bit-Ebene und endet bei der untersten Bit-Ebene

JPEG2000: Enkodierung

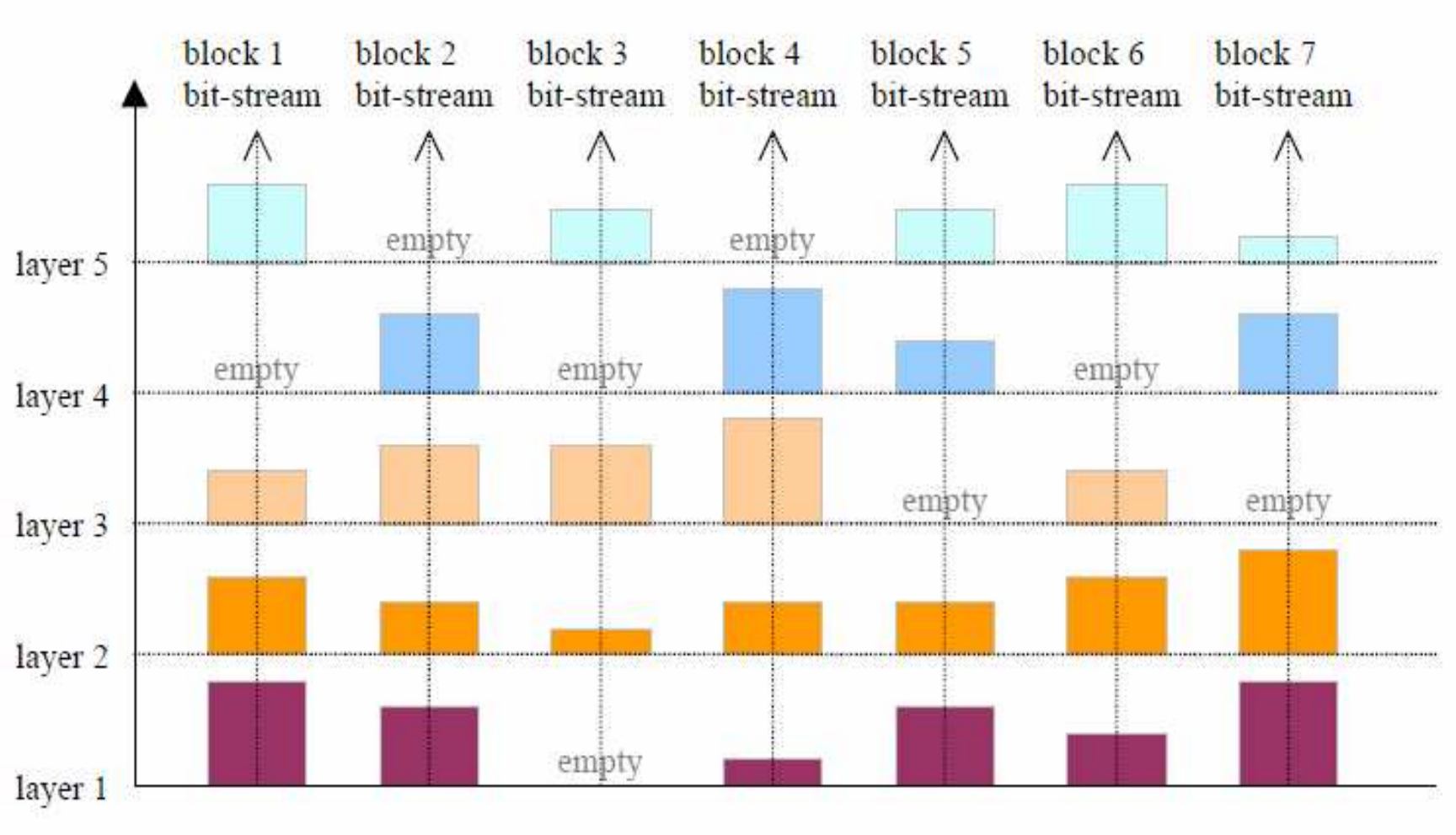
- Die entropiekodierten Daten können dann in einen Bitstrom geschrieben werden
  - Ein Subband nach dem anderen ➔ Progressive by resolution
  - Abwechselnd auf der Coding Unit/Code word Ebene ➔ Progressive by accuracy
  - Beliebig
    - Zufällig
    - Applikationsspezifisch

JPEG2000: Entropiekodierung

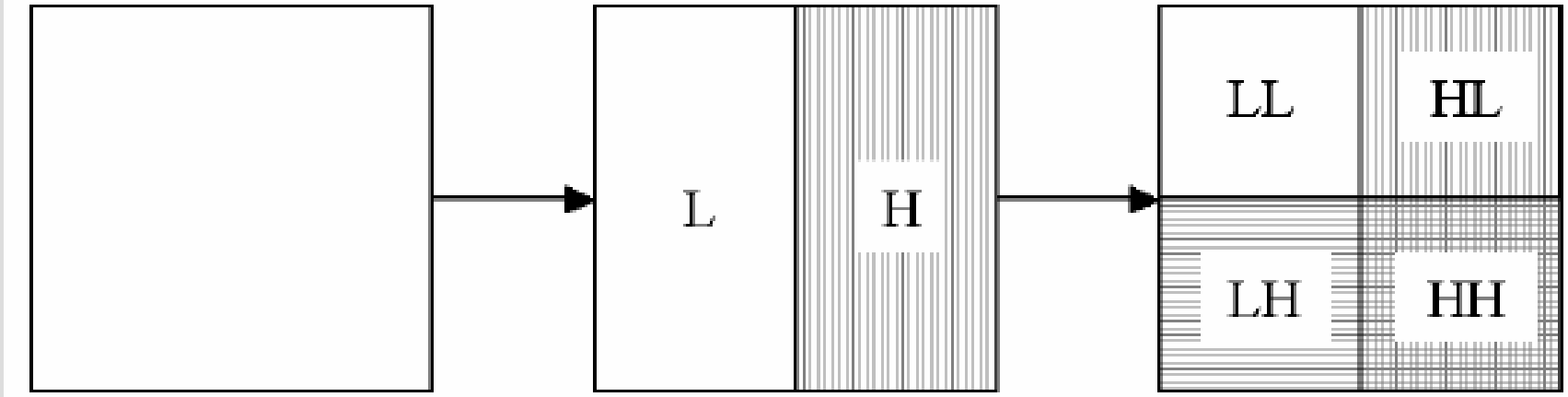
- Pro code block wird also ein Bitstrom erzeugt
  - Dieser Bitstrom kann dann (an best. Positionen) abgeschnitten werden, um der gewünschten Bildqualität zu genügen (z.B. target bit rate)

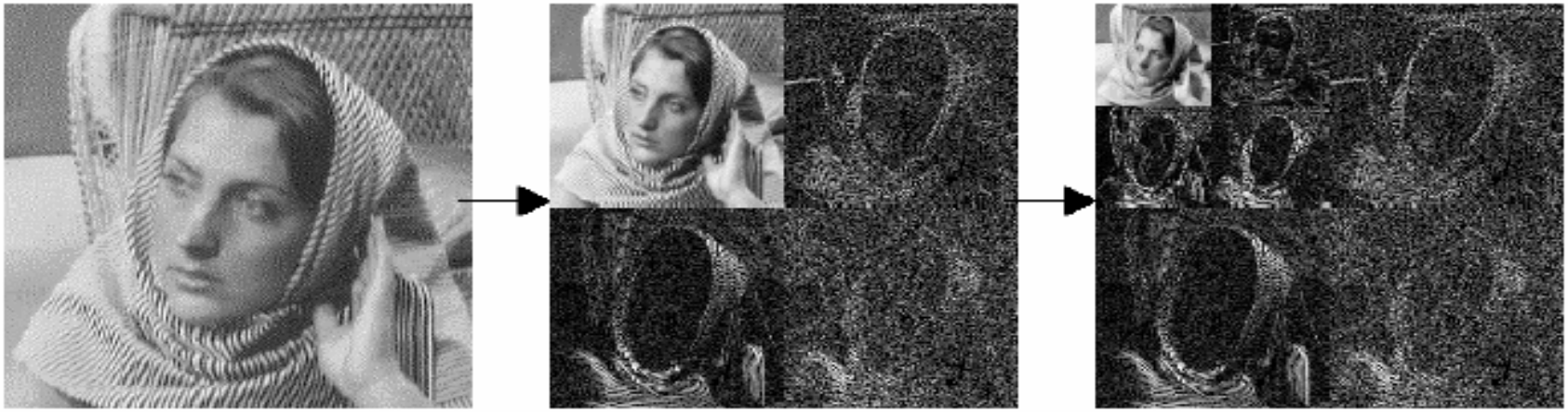


EBCOD Layer Formation



Wavelet Transformation



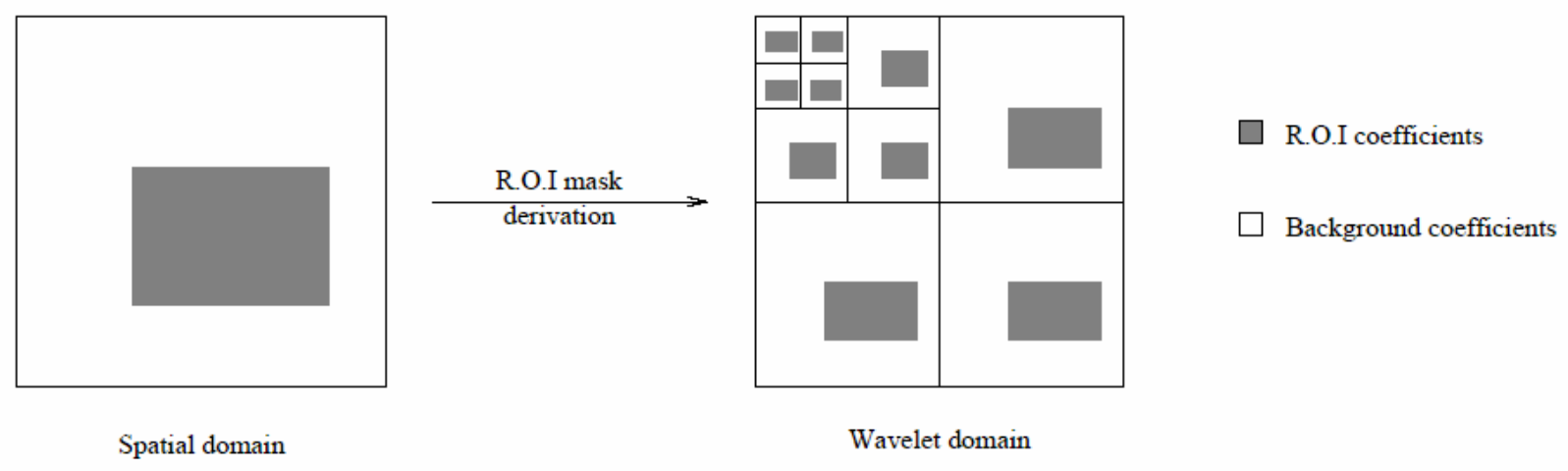


JPEG2000 Regions of Interest

JPEG2000: Regions of Interest

- Definiert in Annex H des JPEG2000 Standards
- Teile des Bildes (ROI) werden mit besserer Qualität kodiert als der Rest (BG, Background)
- Ungleichmäßige Verteilung der Bildqualität
- MaxShift Methode

ROI Kodierung



- Die Wavelet Koeffizienten müssen identifiziert werden
  - Generieren von ROI Masken
- Die Koeff. die zur ROI gehören müssen im restlichen Kodierprozess bevorzugt werden
  - Entweder über eine gröbere Quantisierung der BG-Koeff., oder
  - Verschieben (scaling, shifting) der ROI Koeff. in höhere Bit-Ebenen
    - Diese werden früher kodiert als die Restlichen und erscheinen dementsprechend früher im generierten Bitstrom

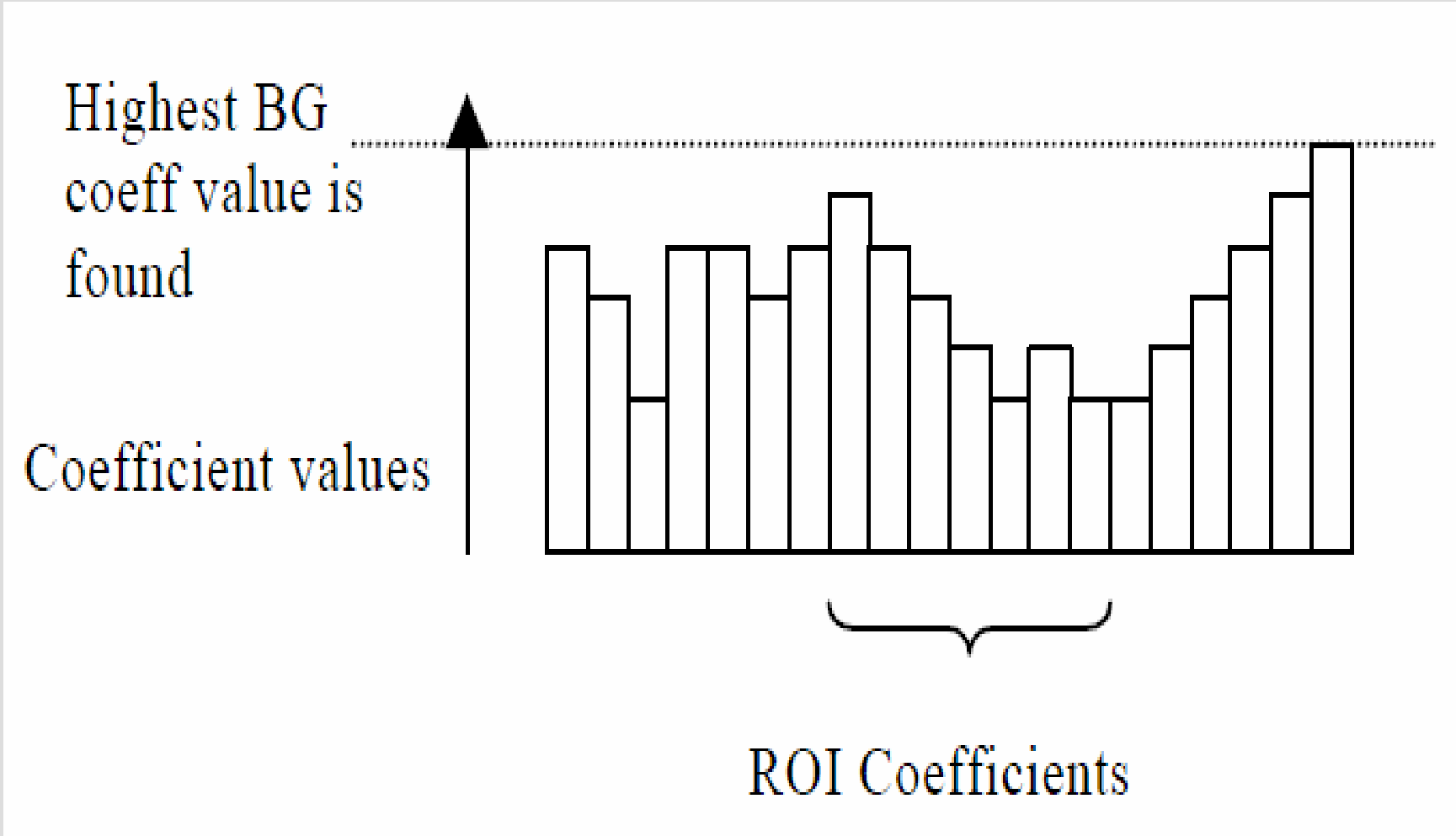
MaxShift Methode (Encoder)

- Der Encoder ermittelt einen Skalierungswert s, sodass der kleinste ROI Koeffizient grösser ist als das Maximum aller BG Koeffizienten.
- D.h., s ist die kleinste Zahl für die gilt:

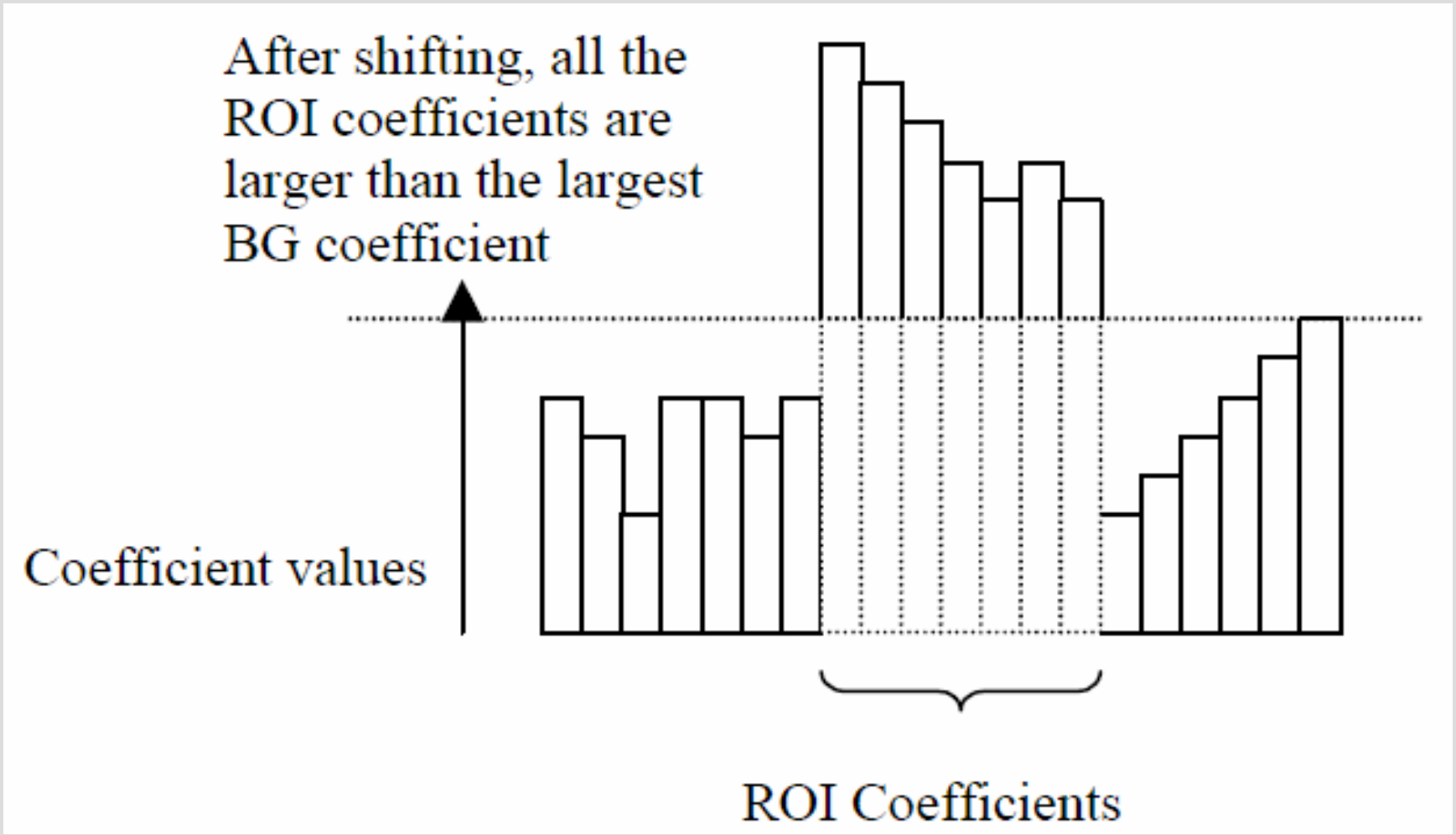
$$\text{Sei } k_{\text{ROImin}} = 2^s$$
$$\forall k_{BG} \in BG : k_{\text{ROImin}} > k_{BG}$$

- Die BG Koeffizienten werden nach unten skaliert
  - Somit wird gewährleistet, dass jeder ROI Koeffizient (!= 0) größer ist als der größte BG Koeffizient
- Der Skalierungswert s wird in den Codestream eingefügt (RGN Marker)

MaxShift Methode



MaxShift Methode



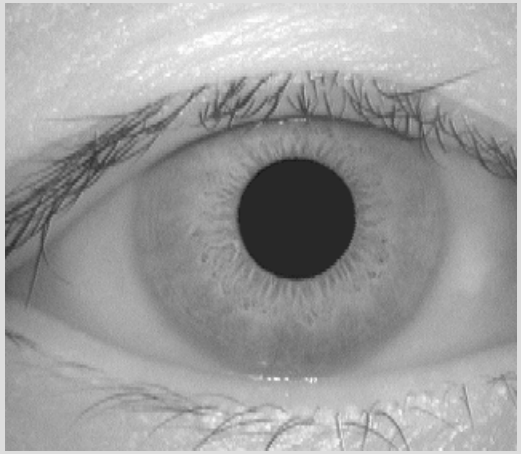
MaxShift Methode (Decoder)

- Wenn der Decoder später den Bitstrom empfängt, extrahiert er zuerst den Skalierungswert s
- Damit kann er ableiten, ob ein Koeffezient k Teil einer ROI ist
  - $k >= k_{ROImin}$ ➔ k ist Teil einer ROI
  - $k < k_{ROImin}$ ➔ k ist Teil des BG
- Er muss nun lediglich die BG Koeffizienten wieder hochskalieren
- Vorteile:
  - Es muss keine zusätzliche Information über die ROIs übertragen werden um das Bild dekodieren zu können.
  - Vereinfacht den Decoder

EyeDROID

(Compress) Eye Data using Region-Of-Interest Detection

Aufgabenstellung



- Bild des Auges einlesen
- Auffinden der IRIS
  - Kantenerkennung/-optimierung
  - Kreiserkennung
- Enkodieren des Bildes mittels JPEG2000 und definierter ROI
- Weiters:
  - Vergleich von Original und JP2000 (mit/ohne ROI Kodierung)
  - Auswirkungen auf die Erkennungsrate (Matching)

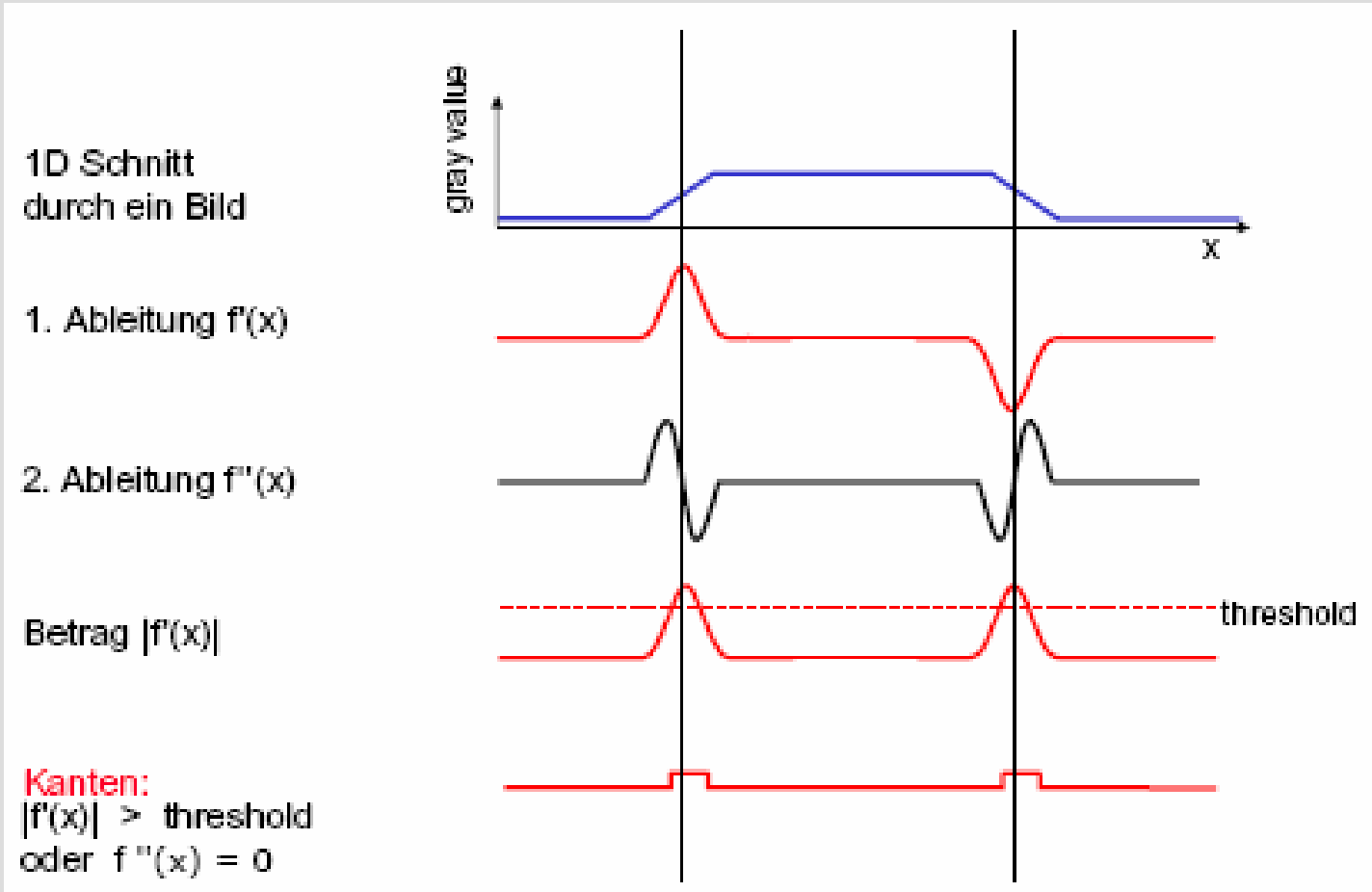
Anforderungen an den Kantenerkennungsalgorithmus

- Genaue Lokalisierung
  - Bei einigen Detektoren kann es zu Verschiebungen kommen. Die Kanten erhalten eine neue Lage oder die Kante „verschmiert“, sie wird z.B. breiter.
- Auf jede Kante sollte nur eine Antwort kommen, um ein Verschmieren der Kante zu verhindern.
- Es sollte vermieden werden, dass Kanten erkannt werden, die tatsächlich nicht vorhanden sind, aber auch, dass tatsächlich vorhandene Kanten nicht gefunden werden (Fehler 1.- und 2.-Art).
- schnelle Berechenbarkeit, geringer Speicherplatzverbrauch.

Canny Edge Detection

- Algorithmus zur Kantenerkennung
- Gliedert sich in verschiedene Faltungsoperationen
- Liefert ein Bild, welches Idealerweise nur noch die Kanten des Ausgangsbildes enthält.
- Arbeitet nur auf Graubildern
- Kanten sind durch große Helligkeitsschwankungen zwischen zwei benachbarten Pixel charakterisiert
  - können somit als Unstetigkeit der Grauwertfunktion $g(x,y)$ des Ausgangsbildes aufgefasst werden

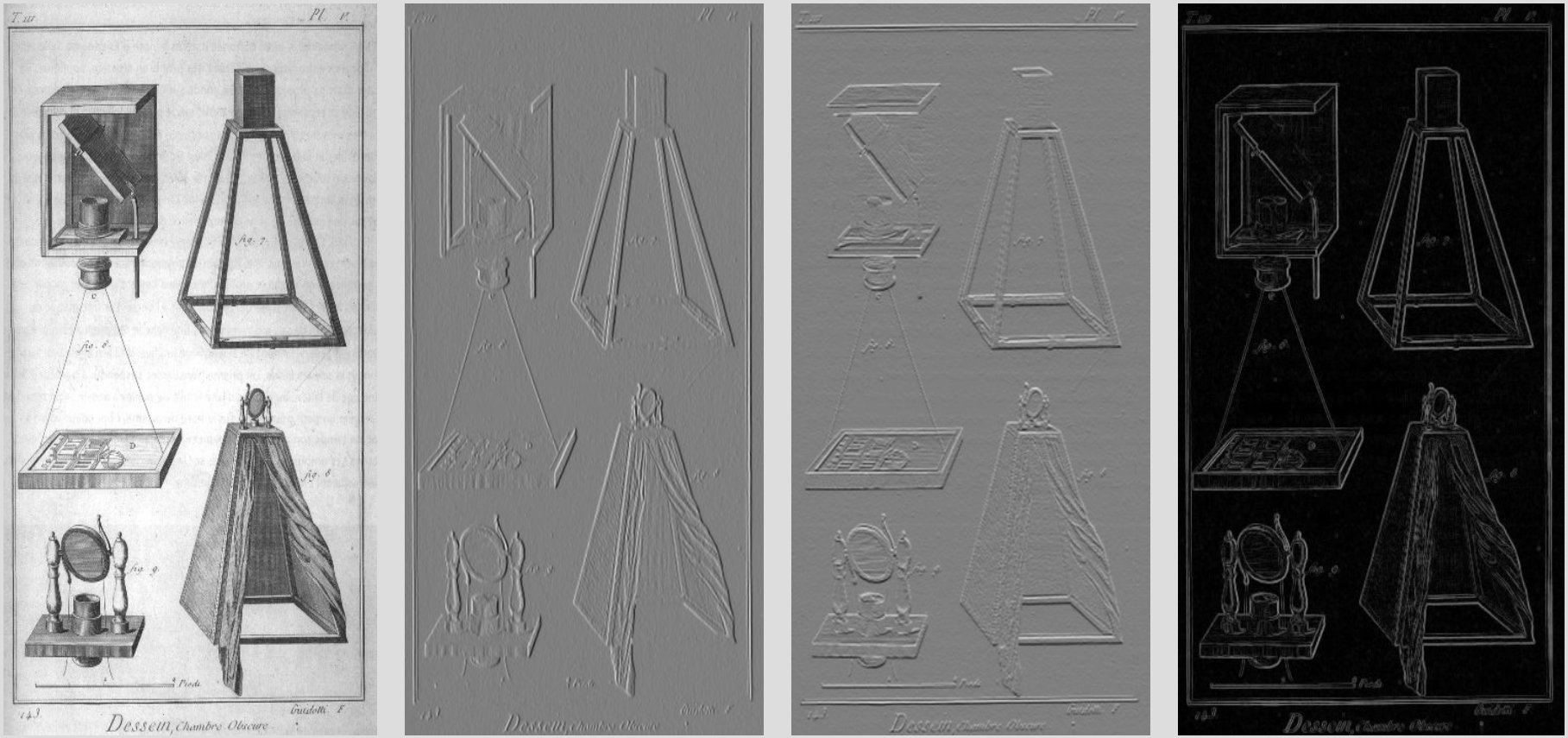
Kantenerkennung



Canny Edge Detection

- Problem: Bildrauschen
  - Gaußsche Normalverteilung zur Glättung des Bildes.
- Gradienten der einzelnen Pixel werden ermittelt, indem das Bild mit Hilfe des Sobeloperators gefaltet wird.
  - **Gradient:** Vektor des größten Anstieges (1. Ableitung).
  - Richtung des größen Helligkeitanstieges.
- Sobeloperator arbeitet entweder in *X*-Richtung oder in *Y*-Richtung und betont somit entweder horizontale oder vertikale Kanten.
- Es ergeben sich also nach Anwendung des Sobeloperators 2 neue Bilder.

Sobel-Operator



Ablauf der Kantenextraktion

1) Gradientenbildung und Sobel-Operationen

2) Non-Maxima-Supression

Kante nur einen Pixel breit: Alle Kantenpixel müssen erkannt werden, die kein maximalen Gradientenbetrag aufweisen -> setzen des Wertes auf 0.

Überprüfung ob einer seiner 8 Nachbarn eine höhere Kantenstärke als der Pixel selbst hat. Errechnete Kantenanstieg von diesem Pixel mit höherer Kantenstärke nicht auf den zu betrachtenden Pixel -> Wert auf 0!

3) Hysterese (Schwellenwertbildung)

Ab welcher Kantenstärke ein Pixel zu einer Kante zu zählen ist.

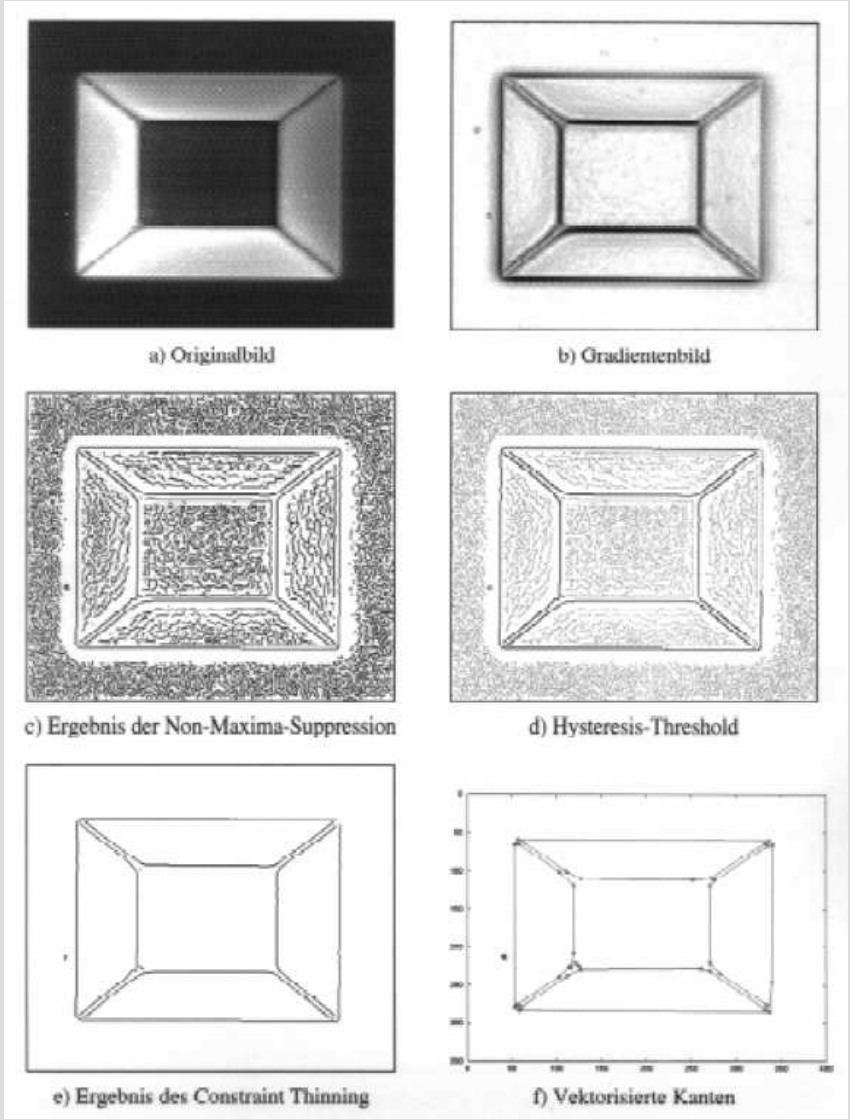
4) Constraint-Thinning-Verfahren

Lücken in den Kanten werden geschlossen. Fehler bei der Non-Maxima-Supression besonders an den Stellen, wo mehrere Kanten aufeinander stoßen.

Man erhält eine Menge von Punkten, die bei korrekter Wahl der Schwellwerte die im Ausgangsbild vorhandenen Kanten aufzeigen.

-> Hough-Transformation

Ablauf der Kantenextraktion



Hough Circle Detection

- Verfahren zur Erkennung von Kreisen
- Auffinden von Kreisen, die durch die Formel

$$(x - x_0)^2 + (y - y_0)^2 = r_0^2$$

  beschrieben werden.
- Ausgangspunkt (x_0,y_0) des Bildes
  - Summe aller Grauwerte der Punkte (x, y), die um den Radius r_0 von (x_0,y_0) entfernt sind und die Kreisgleichung erfüllen, in ein dreidimensionales accumulator array an der Stelle (x_0,y_0,r_0) eingetragen.
- Für jede mögliche Kombination - alle Pixel durchlaufen und die Erfüllung der Kreisgleichung überprüfen.
- Der Eintrag im Ergebnisraum mit dem maximalen Wert kennzeichnet die Parameter des im Bild gesuchten Kreises.

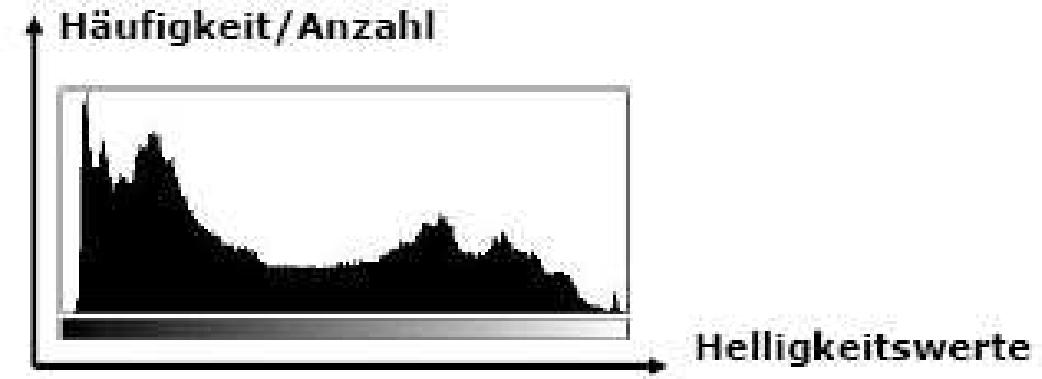
Hough Circle Detection

- Oft gibt es nur sehr wenige Punkte, die dem Kreis zugeordnet werden können

$$(x - x_0)^2 + (y - y_0)^2 - {r_0}^2 \leq \varepsilon \quad \varepsilon \in \mathbb{R}$$

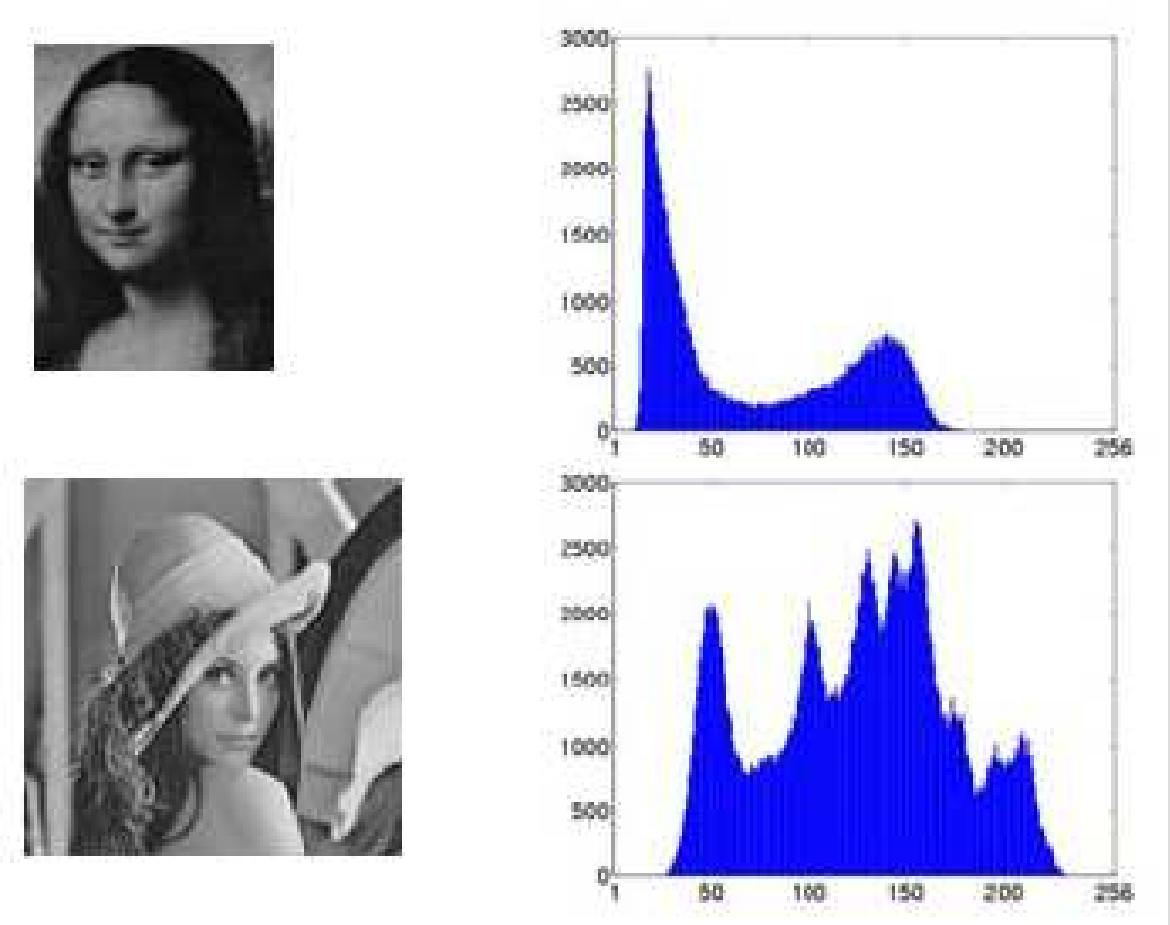
- *Epsilon* muss so gewählt werden, dass genügend Punkte des Kreises die Gleichung erfüllen und somit in das Accumulator Array aufgenommen werden.
- Ein zu großer Wert würde zu viele Pixel dem Kreis zuordnen, ein zu kleiner Wert zu wenige.

Histogramme



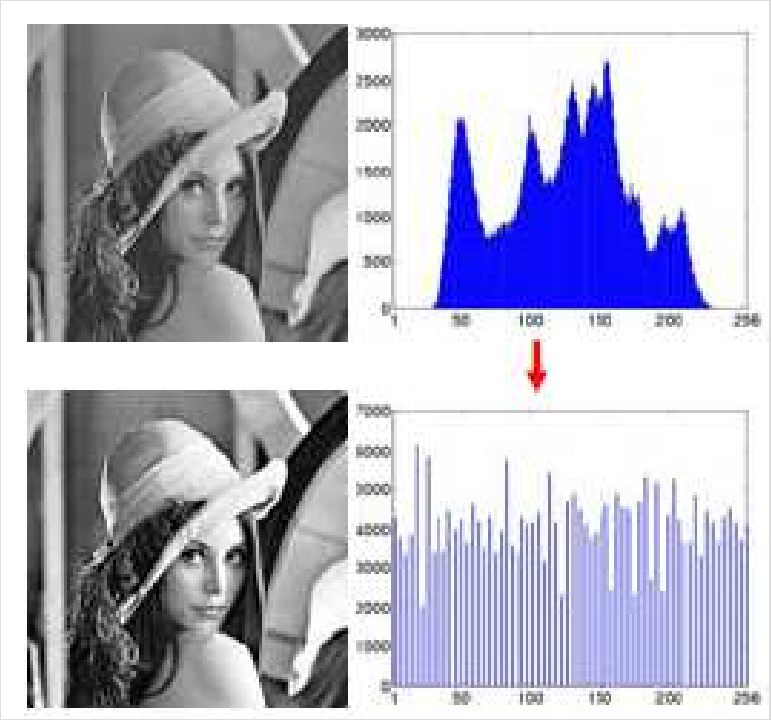
- **Darstellung der Häufigkeit aller Werte im Bild** geordnet nach Helligkeit.
- Die Position spielt keine Rolle.

Beispiele


3000
2500
2000
1500
1000
500
0
1
50
100
150
200
256
3000
2500
2000
1500
1000
500
0
1
50
100
150
200
256

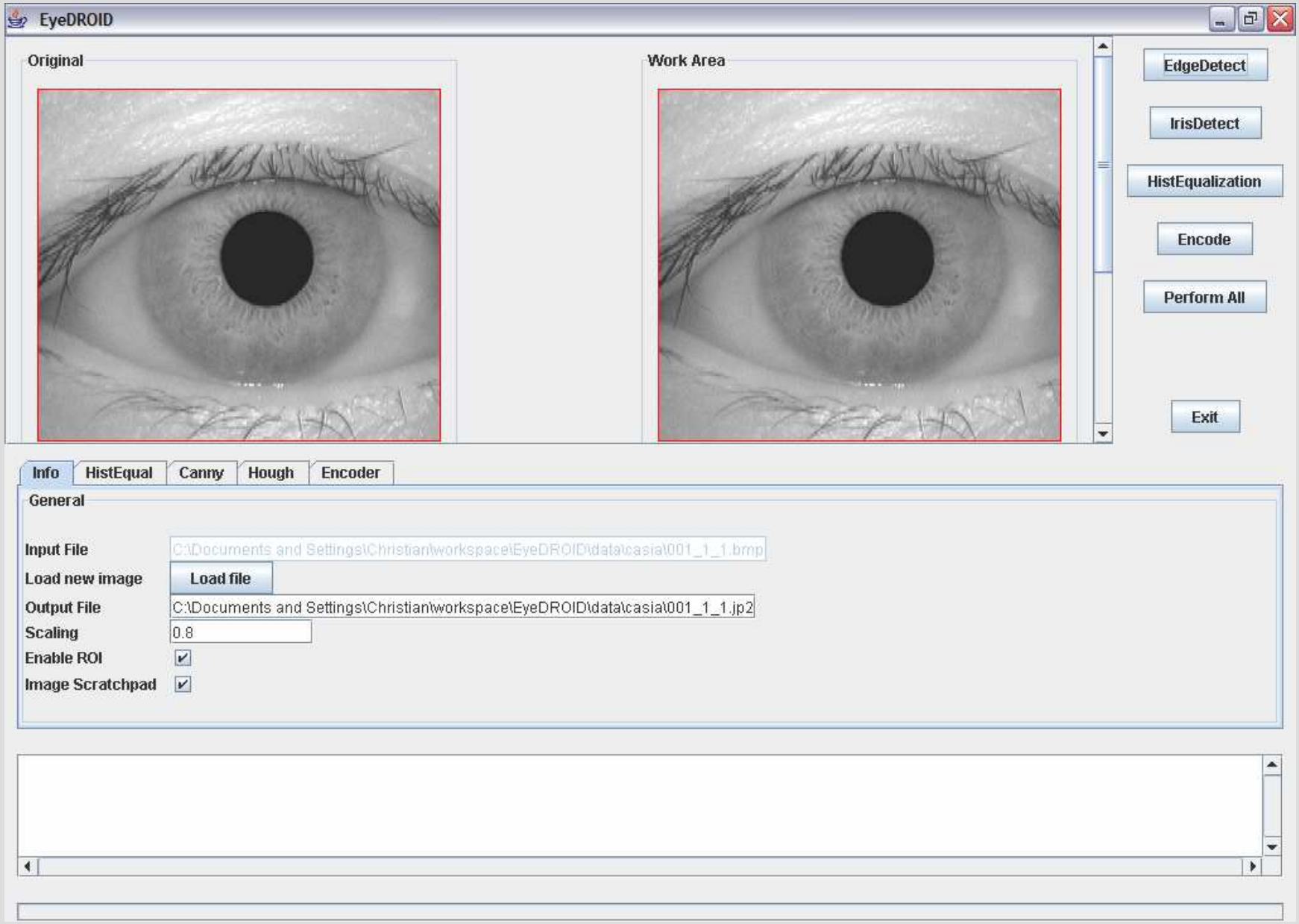
Histogram equilization

- Histogrammeinebnung
- Gleiche Anzahl von Pixeln pro Grauwertintervall
- alle Grauwerte kommen (ungefähr) gleichhäufig vor:
  - höherer Bildkontrast



EyeDROID

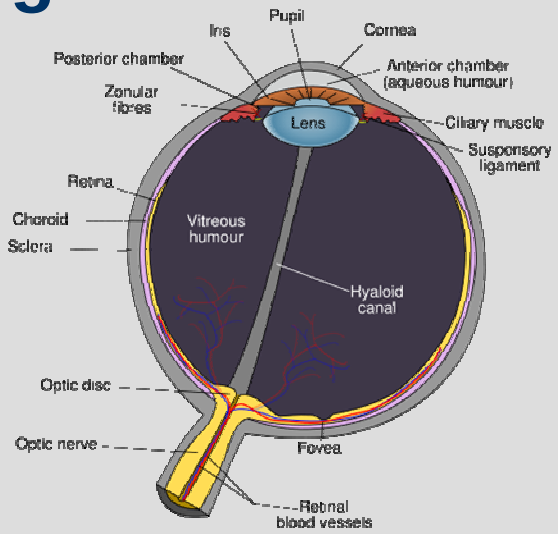
- Implementiert in Java
- JJ2000 für JPEG2000 Enkodierung
  - Java-Referenzimplementierung von JPEG2000

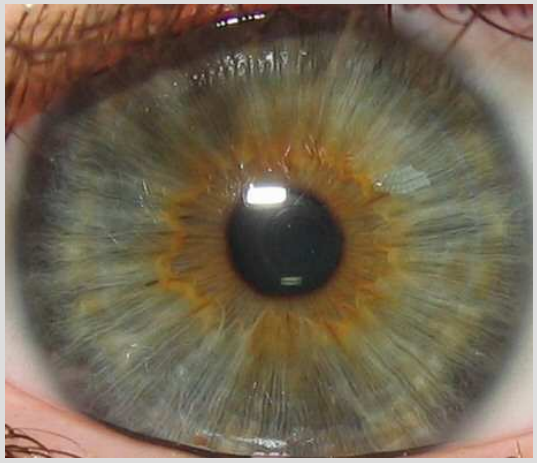

EyeDROID
Original
Work Area
EdgeDetect
IrisDetect
HistEqualization
Encode
Perform All
Exit
Info
HistEqual
Canny
Hough
Encoder
General
Input File
C:\Documents and Settings\Christian\workspace\EyeDROID\data\casia\001_1_1.bmp
Load new image
Load file
Output File
C:\Documents and Settings\Christian\workspace\EyeDROID\data\casia\001_1_1.jp2
Scaling
0.8
Enable ROI
Image Scratchpad

Iriserkennung

Iriserkennung

- Sehr verläßliches biometrisches Verfahren
  - Dir Irismuster entwickeln sich zufällig (genetische Faktoren haben keinen Einfluß)
  - Irismuster bildet sich sehr früh und bleibt das Leben über erhalten
- Durchschn. Irisgrösse: 12mm
- Größe der Pupille: 10% - 80% vom Durchmesser der Iris





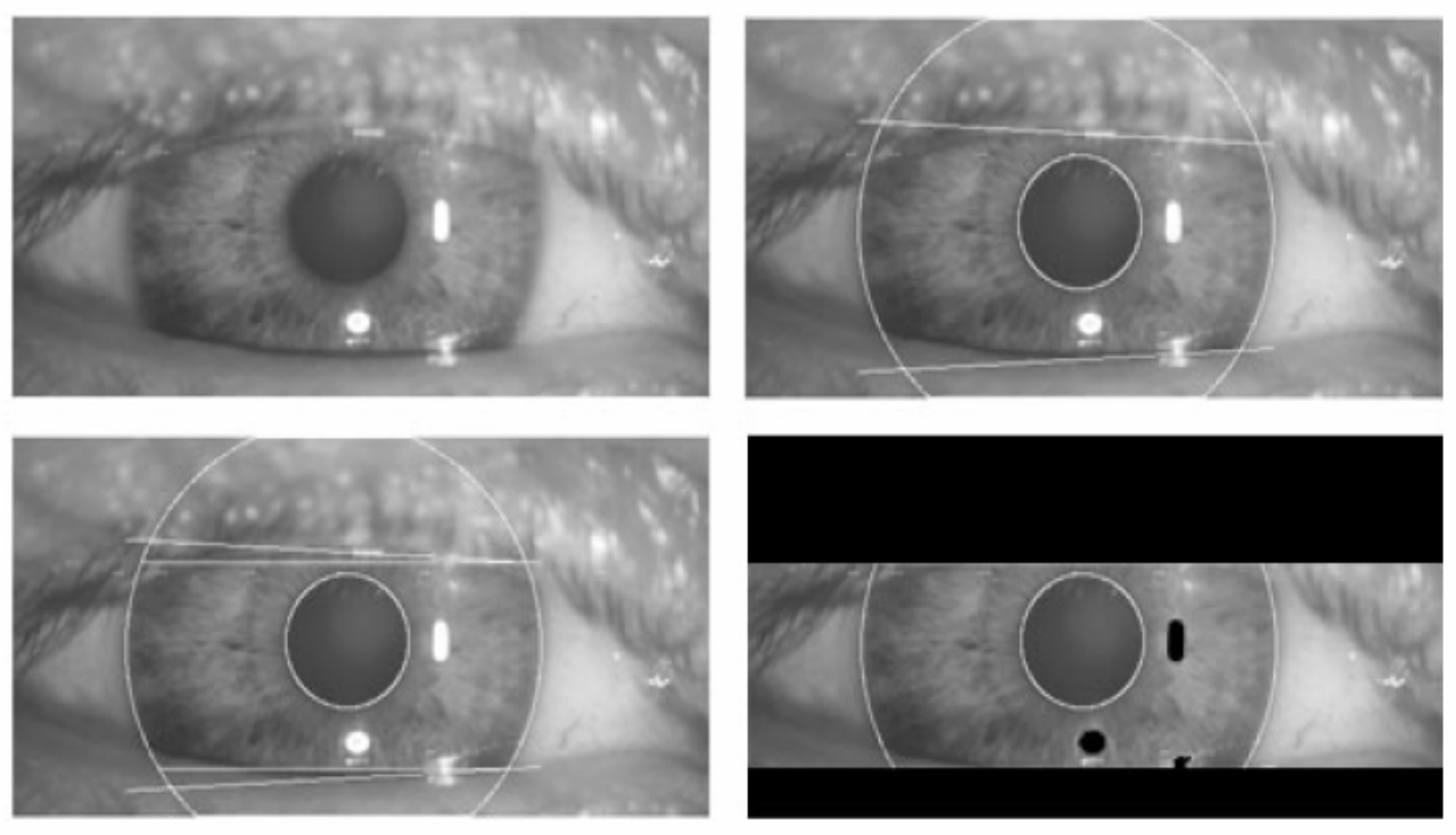
Iris Recognition System (IRS)

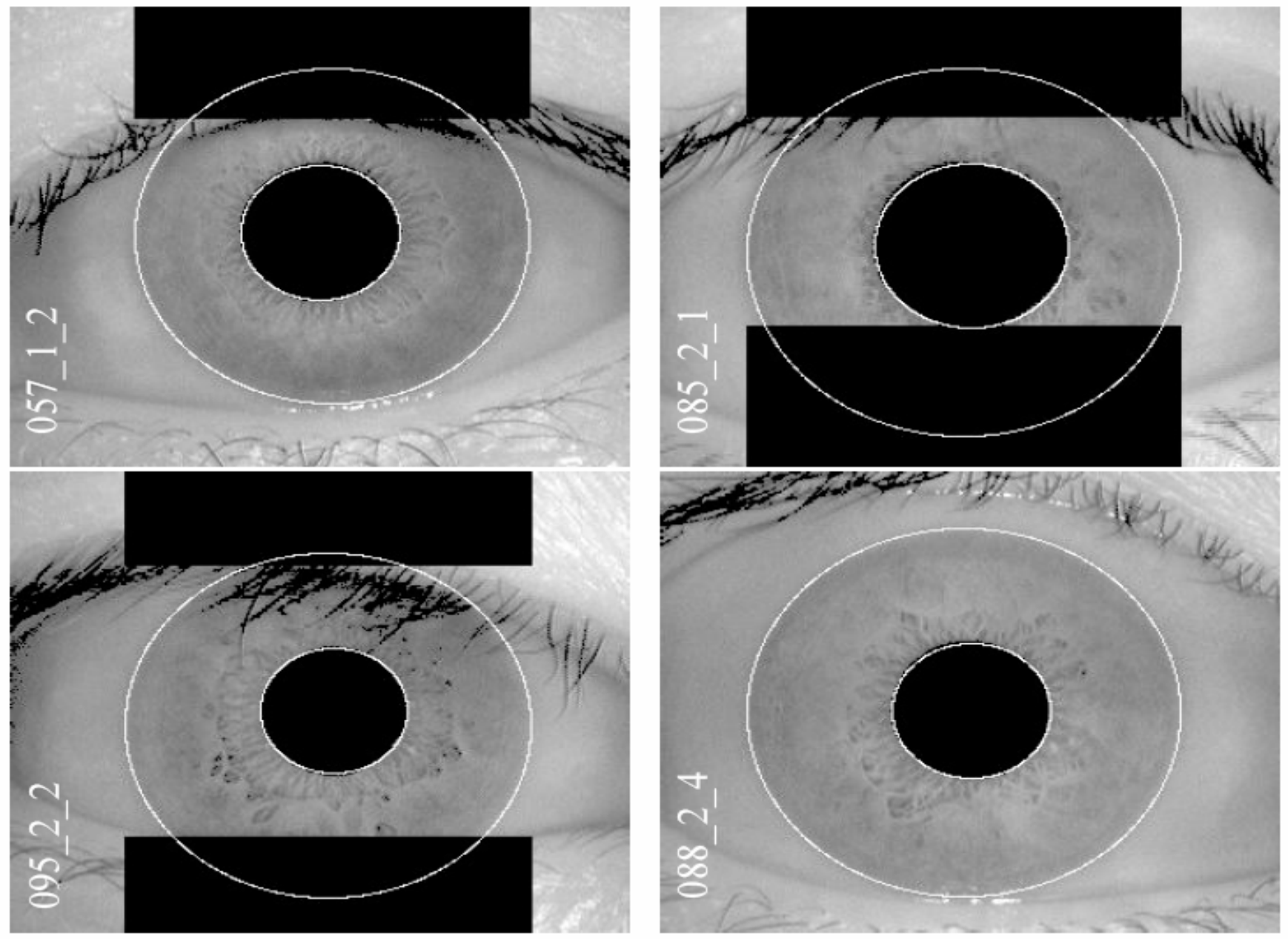
- Die meisten Systeme zur Iriserkennung nutzen patentierte Algorithmen (z.B. Daugman) und sind nicht frei verfügbar
- Matlab SW von Libor Masek
  - “Open-Source” Skriptsammlung für Matlab mit der man Iriserkennung betreiben kann
  - Im Rahmen der Dissertation von Libor Masek entstanden (an der University of Western Australia)
  - Frei verfügbar für Forschungs- und Testzwecke
  - http://www.csse.uwa.edu.au/~pk/studentprojects/libor/sourcecode.html

IRS: Biometrische Vorlage

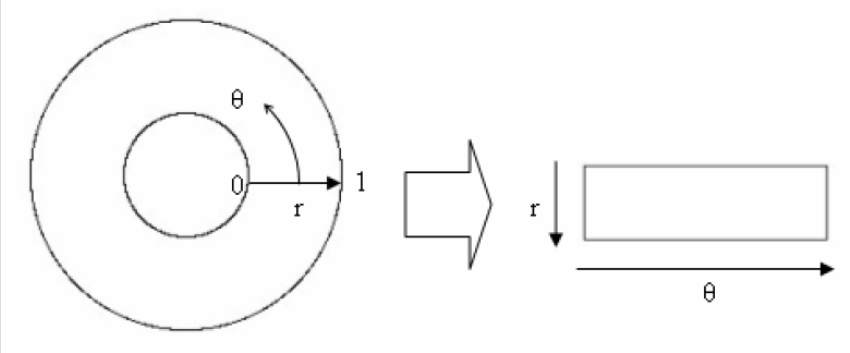
1. Automatische Segmentierung der Iris basierend auf
   - Canny Edge Detection zur Kantenfindung und
   - Hough Transformation zur Erkennung von Kreisstrukturen
2. Erkennung von
   - Augenlidern
   - Wimpern
   - Reflektionen
3. Irisregion wird normalisiert ➔ Irisband
   - Rechteckiger Bereich mit konstanter Grösse
4. Phaseninformation eines 1D Log-Gabor (Wavelet) Filters wird quantisiert (4 Stufen)
   - Irismuster wird in einer biometrischen Maske/Vorlage gespeichert (template)

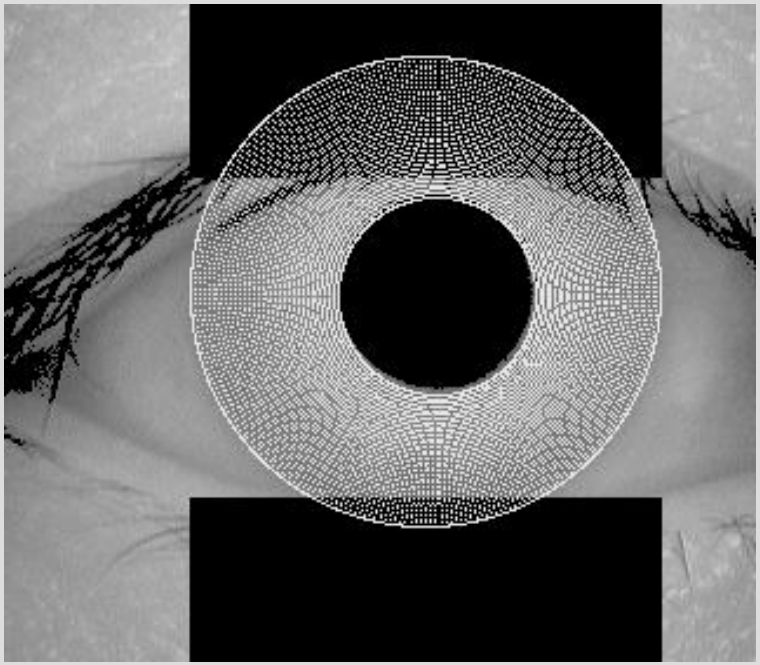
Entfernen von Augenlidern




057_1_2
085_2_1
095_2_2
088_2_4

IRS: Daugman's Rubber Sheet Model





- Algorithmus für die Normalisierung
- r … Radiale Auflösung
- Theta … Winkelauflösung

Irisband

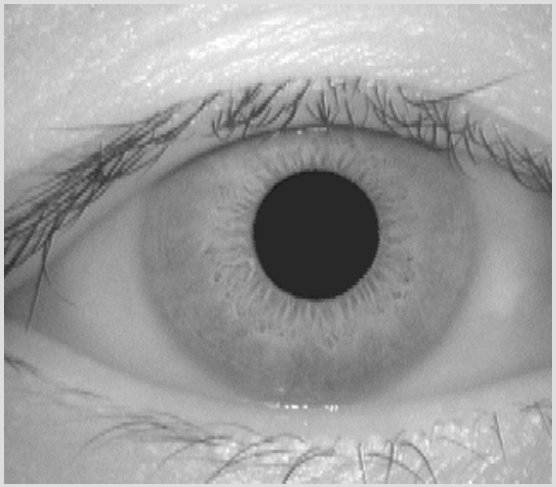
IRS: Iris Vergleich (Matching)

- Mittels Hamming-Distanz
  - Gibt an, wie sehr sich zwei Bitmuster unterscheiden
- Für zwei unterschiedliche Templates sollte die Hamming-Distanz (HD) 0.5 sein (unabhängig, zufällig)
- Für zwei Templates der selben Iris sollte die HD nahe 0 sein (hohe Korrelation)
- Ausgleich rotationeller Unterschiede
  - Niedrigste HD wird zurückgeliefert

Resultate

Irisbilder

- CASIA
  - Chinese Academy of Sciences – Institute of Automation
  - 320 x 280 (=89600) Pixel, 8bpp, Bitmap
  - Nahe-Infrarot Aufnahmen
  - Radien
    - Iris: 90 bis 150 pixel
    - Pupille: 28 bis 75 pixel



- UBIRIS
  - 200 x 150 (=30000) Pixel, 24bpp, Color JPEG

CASIA Tests

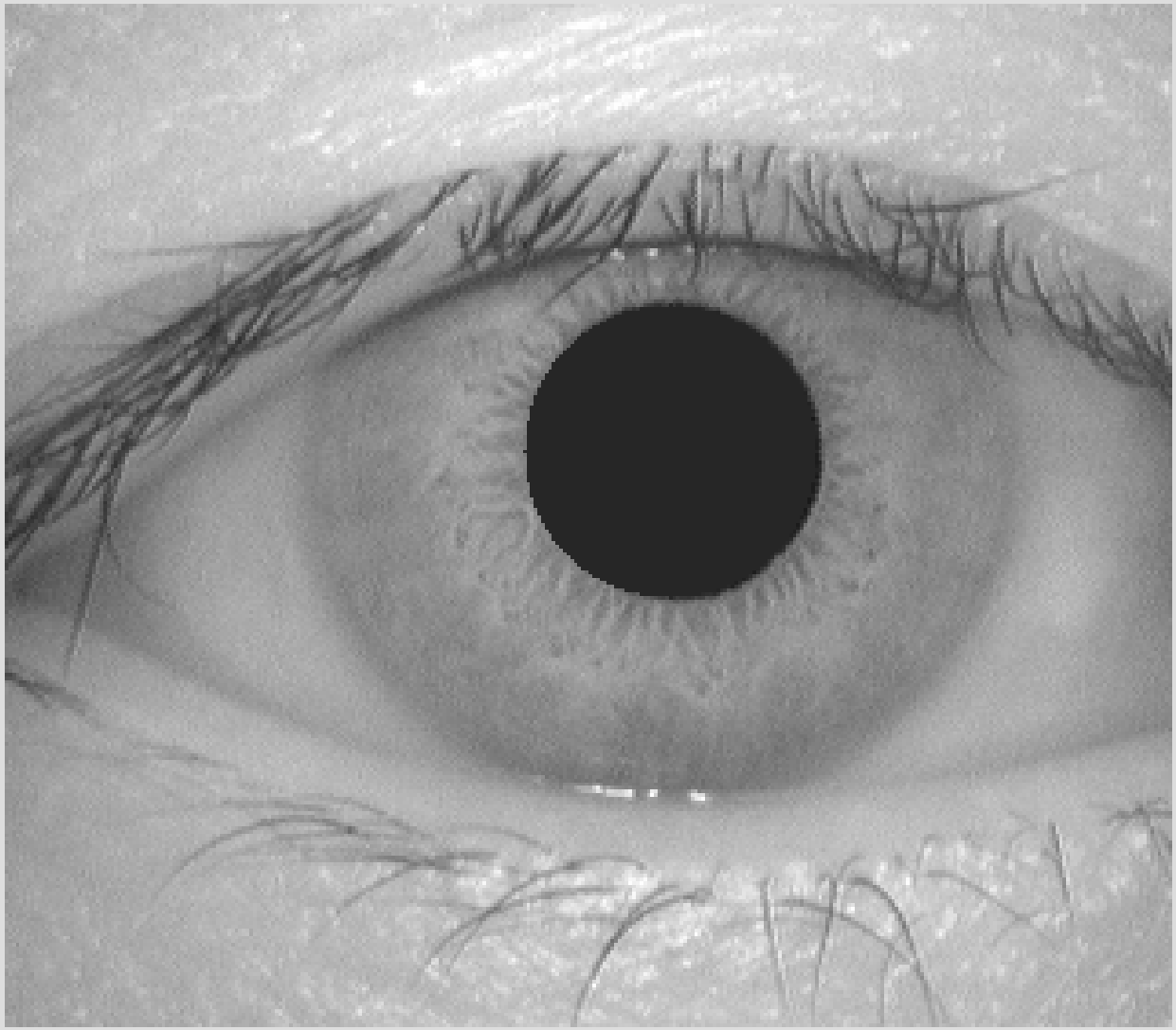
- Pro Person 4 (untersch.) Aufnahmen der Iris ➔ Image Set
  - 1. Bild wird als Referenzbild angenommen
  - Die verbleibenden 3 Irisbilder werden jeweils als
    - Original
    - JPEG2000 (J2K)
    - JPEG2000 mit ROI (J2KwROI)

    mit dem Referenzbild verglichen
  - Als Ergebnis wird dann jeweils der Durchschnitt der 3 Vergleiche angenommen
- Insgesamt 20 Image Sets
- Ein Image Set benötigt etwa 17 Minuten, bei 3 untersch. Bitraten
  System: 3 GHz Pentium 4 (1MB L2 Cache), 1GB Hauptspeicher

CASIA Tests

- Separation bei HD=0.4
  - < 0.4 ➔ Übereinstimmung
  - > 0.4 ➔ keine Übereinstimmung
- Radiale Auflösung r=32
- Winkelaufösung Theta=280

CASIA 001_2_1 (Original)

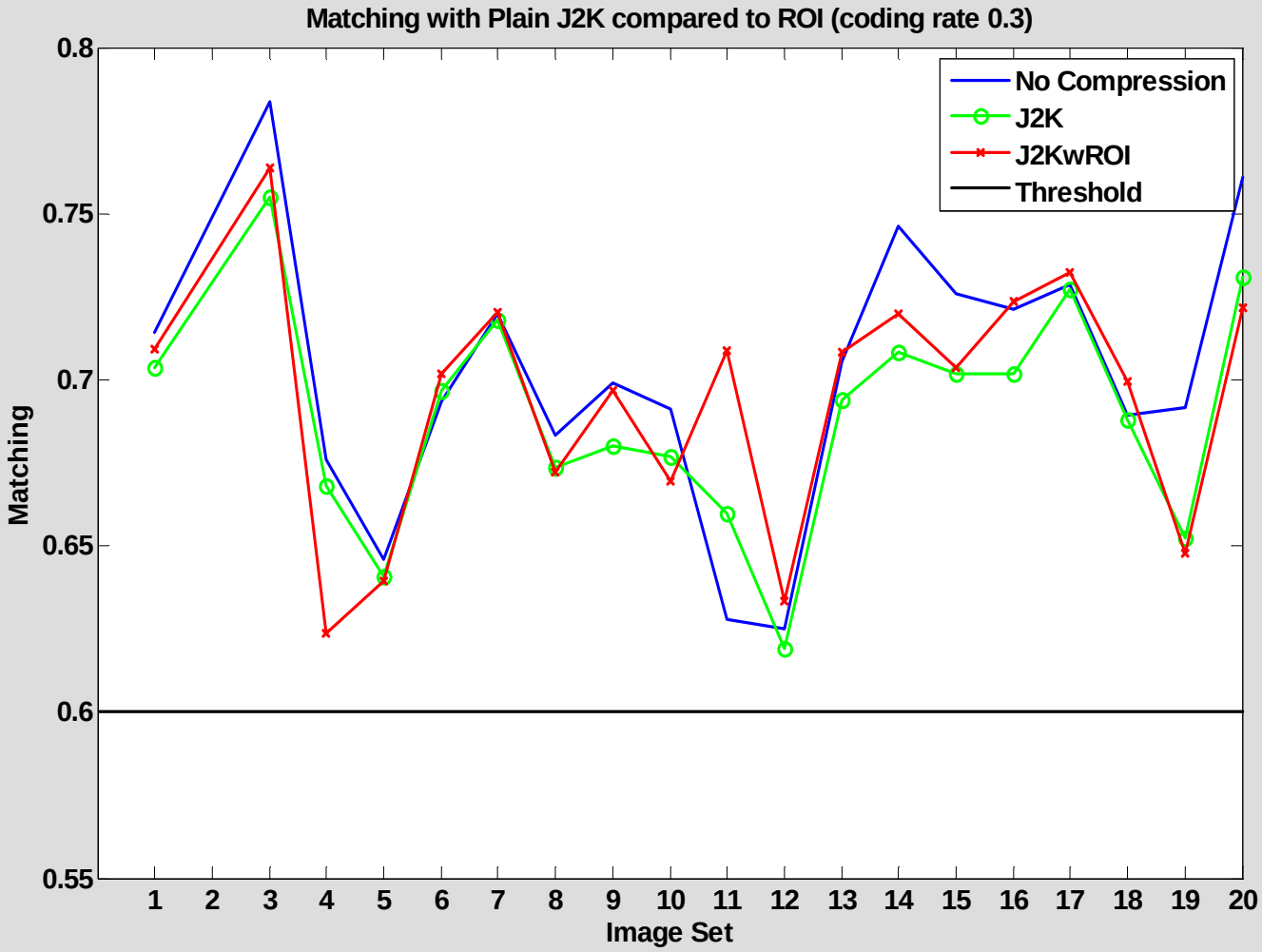


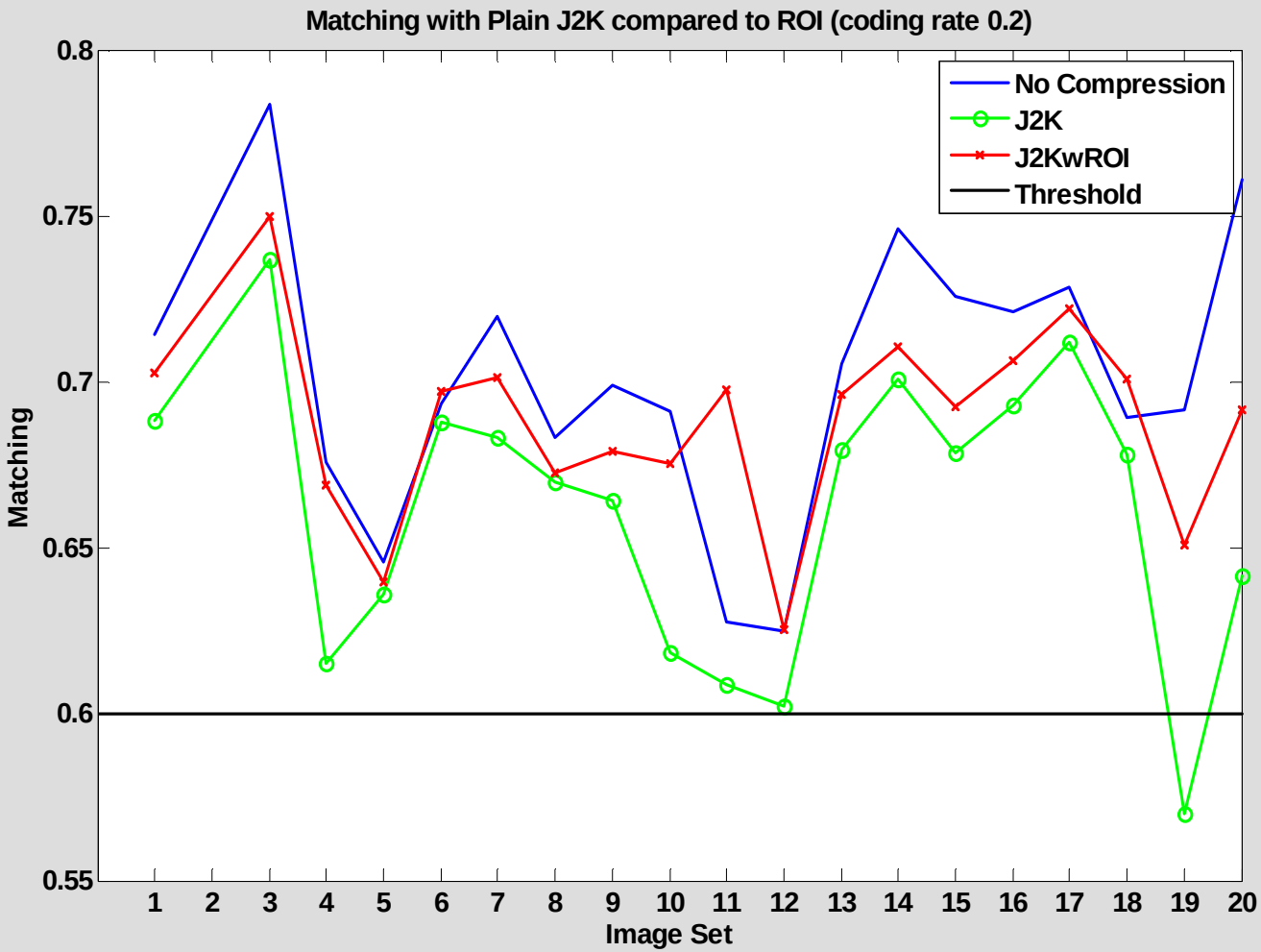
CASIA 001_2_1 (J2K @ 0.2 bpp)

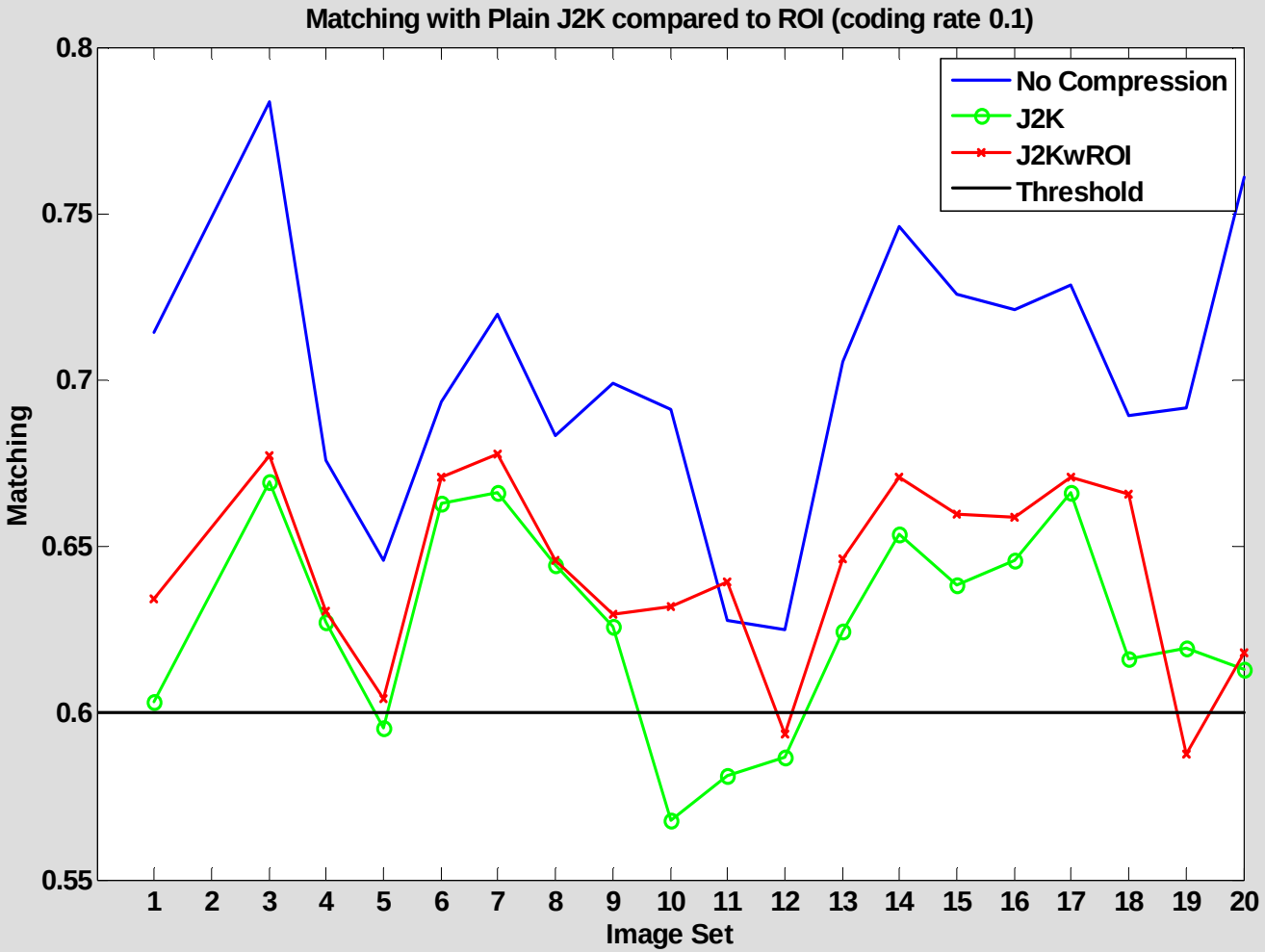
CASIA 001_2_1 (J2KwROI @ 0.2 bpp)

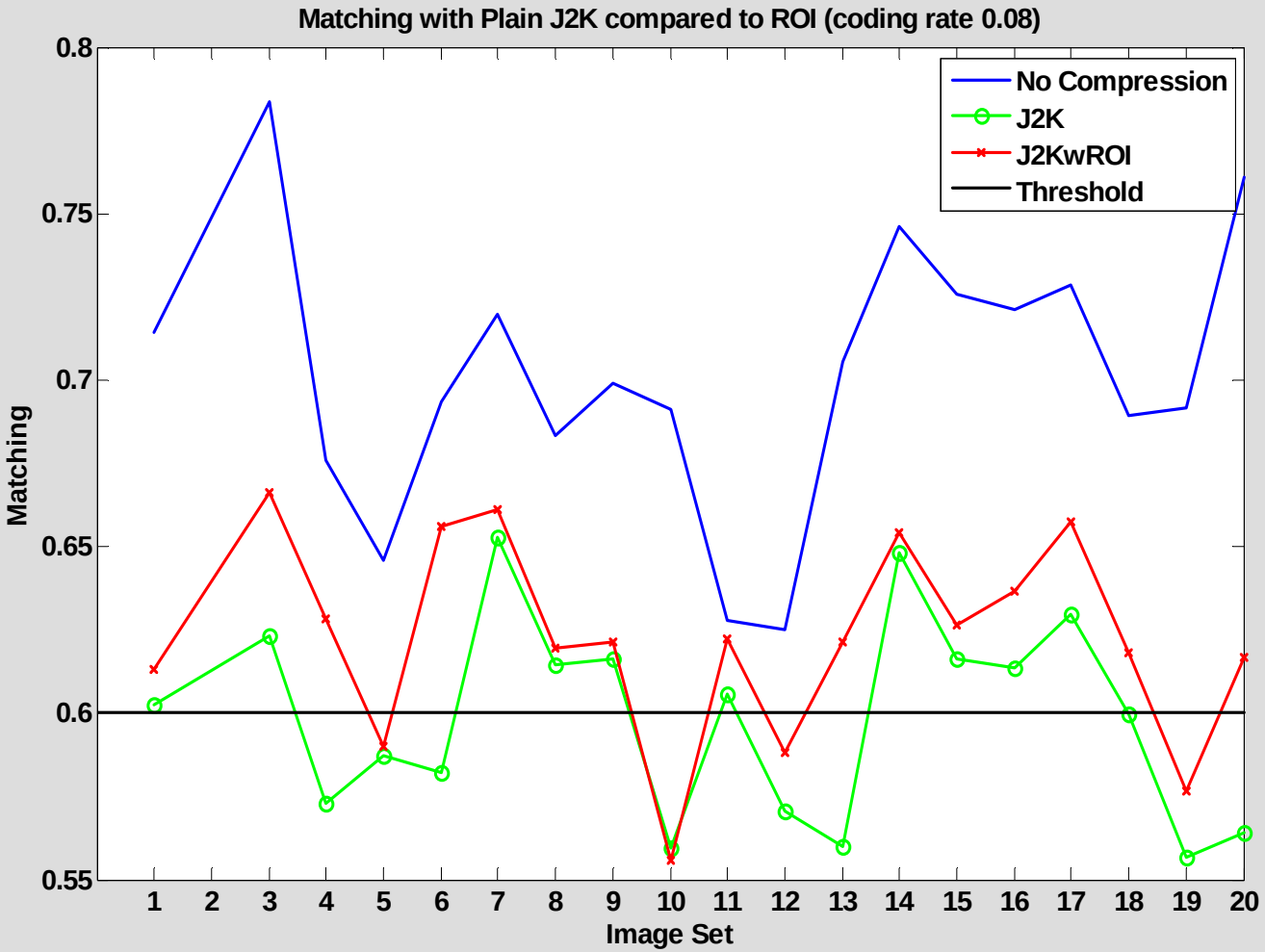
Resultate

- Region-of-Interest
  - ⌀ Radius: 106.0375 Pixel
  - ⌀ Größe: 35416 Pixel (Gesamtbild 89600)
  - ⌀ Größe relativ zum Gesamtbild: 40 %

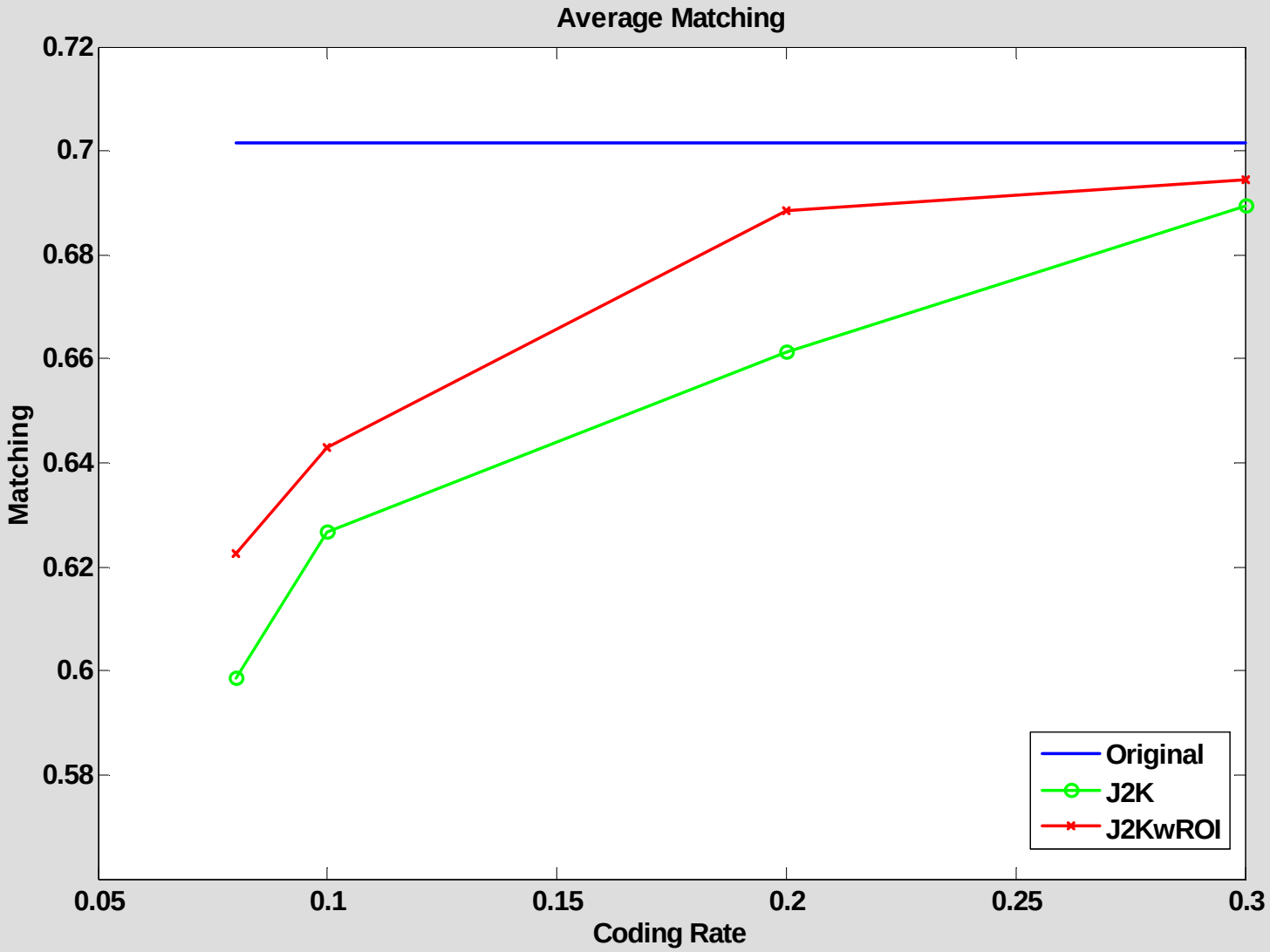

Matching with Plain J2K compared to ROI (coding rate 0.3)
No Compression
J2K
J2KwROI
Threshold
Matching
Image Set
0.8
0.75
0.7
0.65
0.6
0.55
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20


Matching with Plain J2K compared to ROI (coding rate 0.2)
No Compression
J2K
J2KwROI
Threshold
Matching
Image Set
0.8
0.75
0.7
0.65
0.6
0.55
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

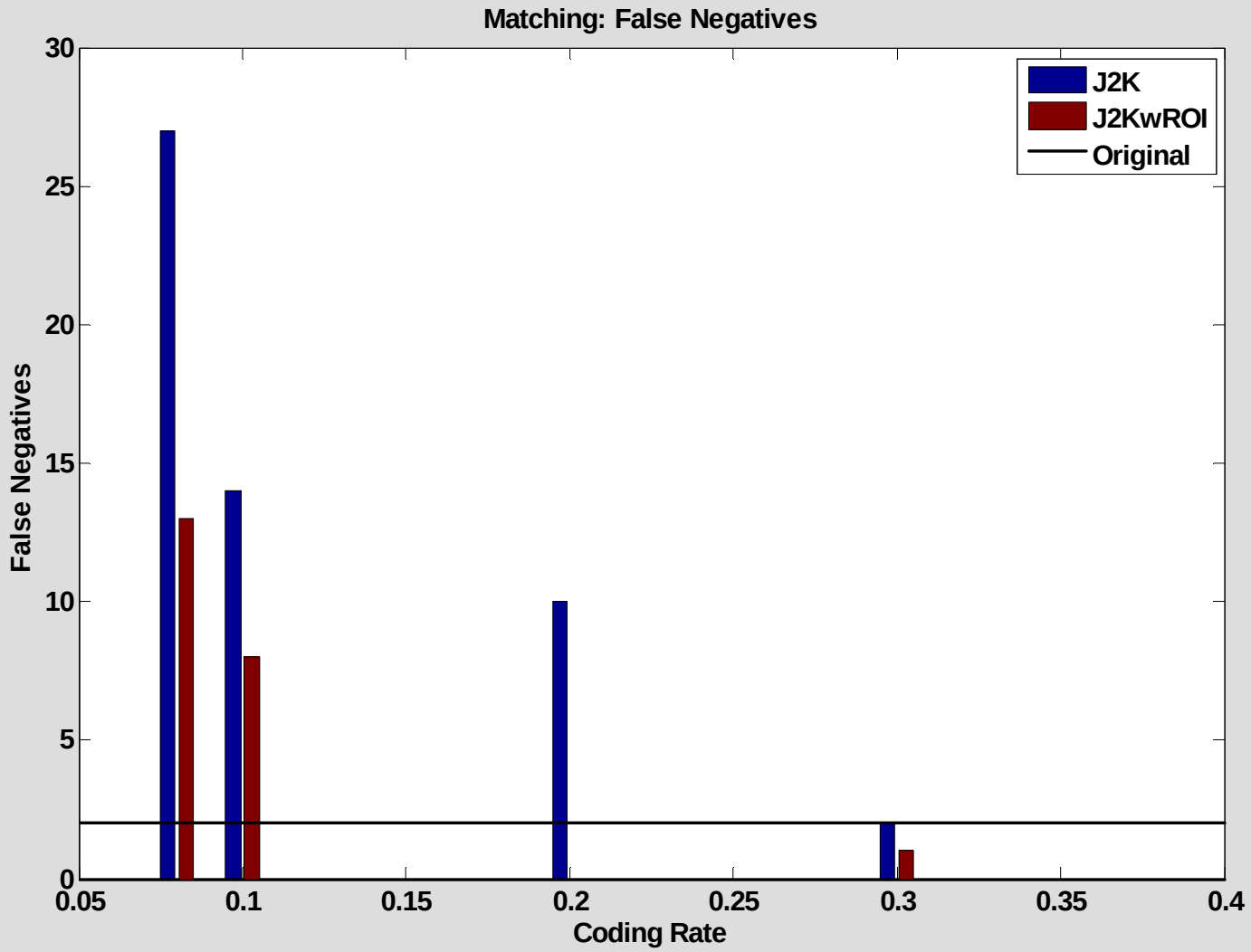

Matching with Plain J2K compared to ROI (coding rate 0.1)
No Compression
J2K
J2KwROI
Threshold
Matching
Image Set
0.8
0.75
0.7
0.65
0.6
0.55
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20


Matching with Plain J2K compared to ROI (coding rate 0.08)
No Compression
J2K
J2KwROI
Threshold
Matching
Image Set
0.8
0.75
0.7
0.65
0.6
0.55
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

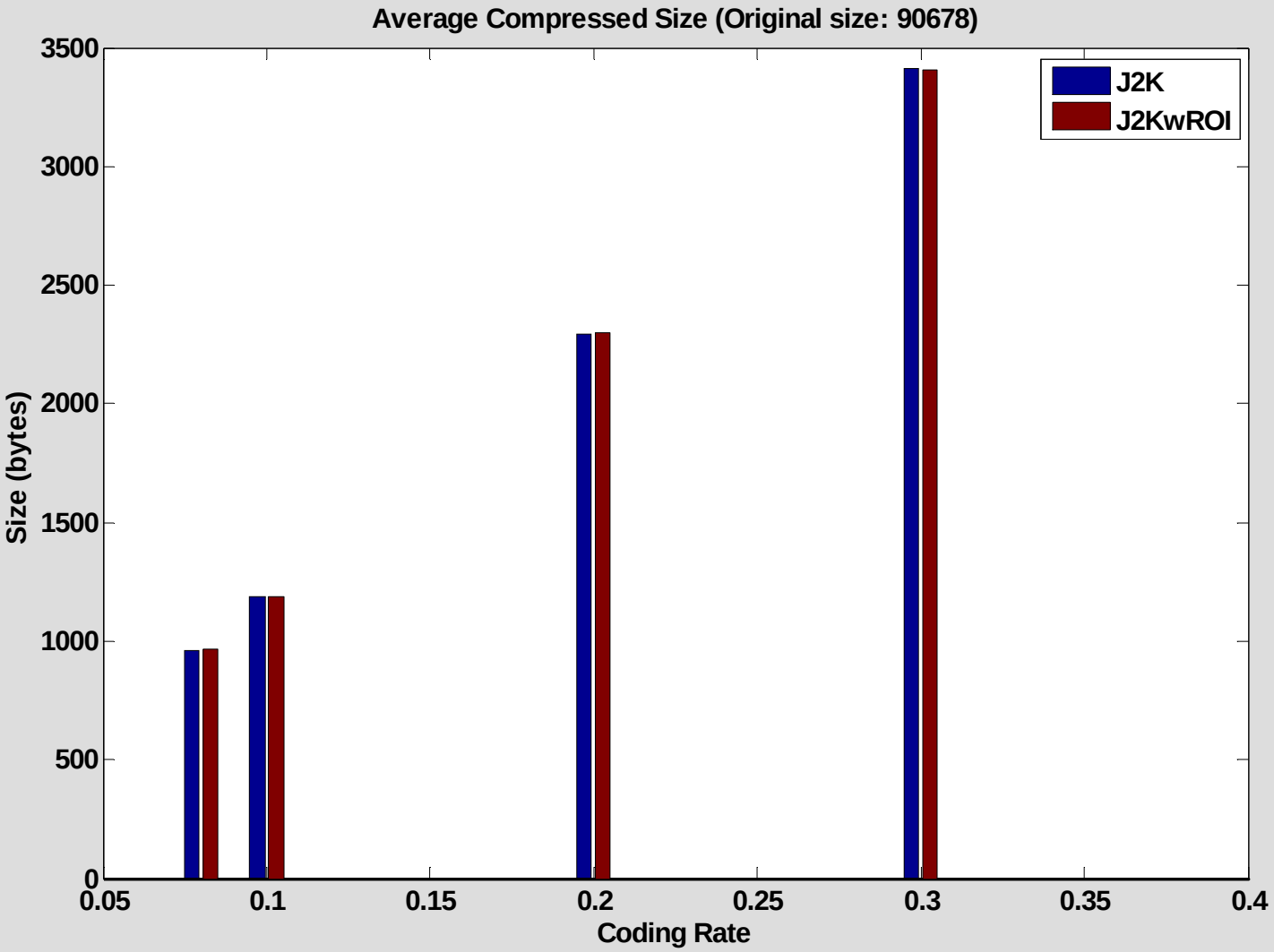
CASIA: Matching Performance

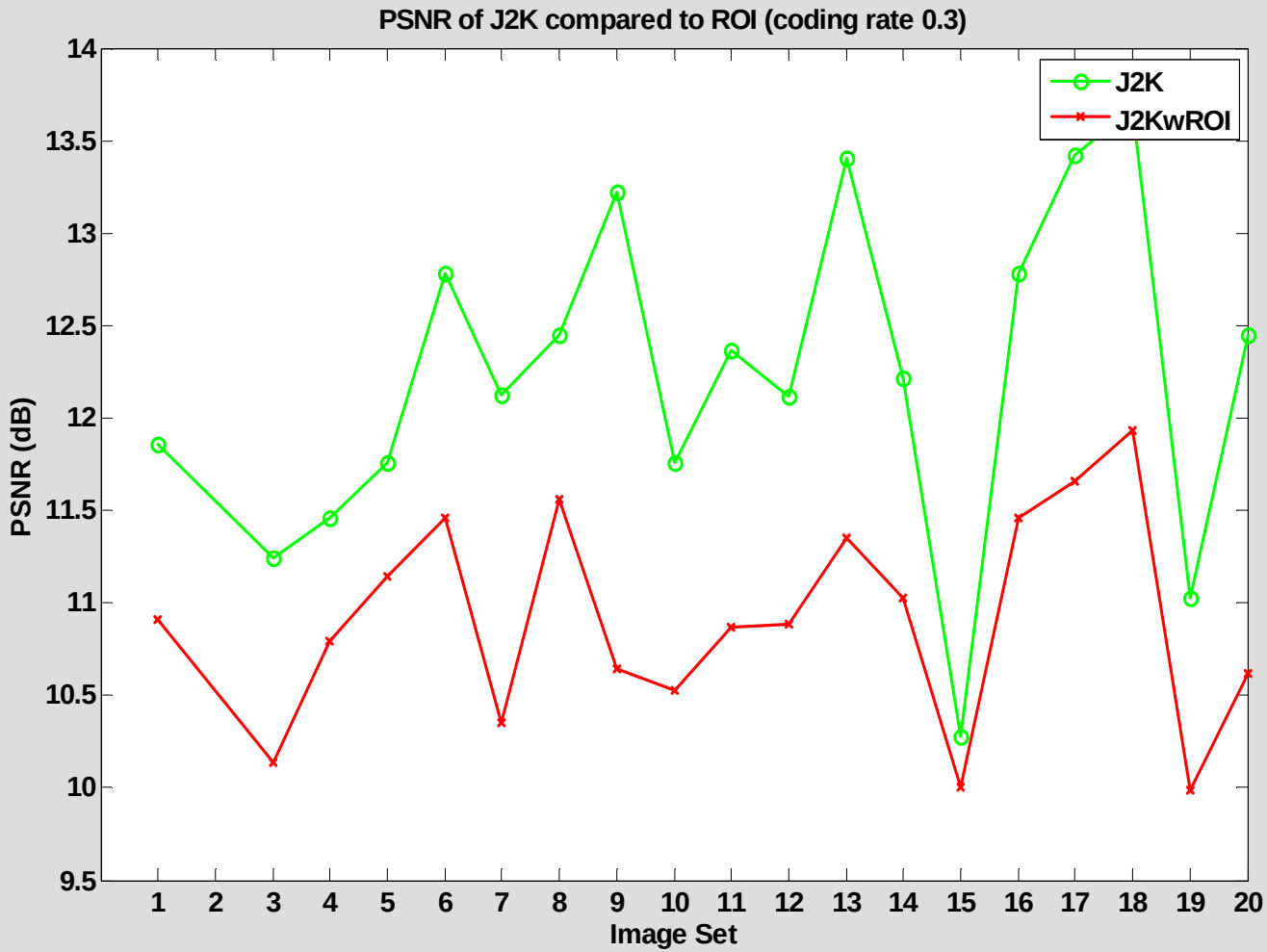


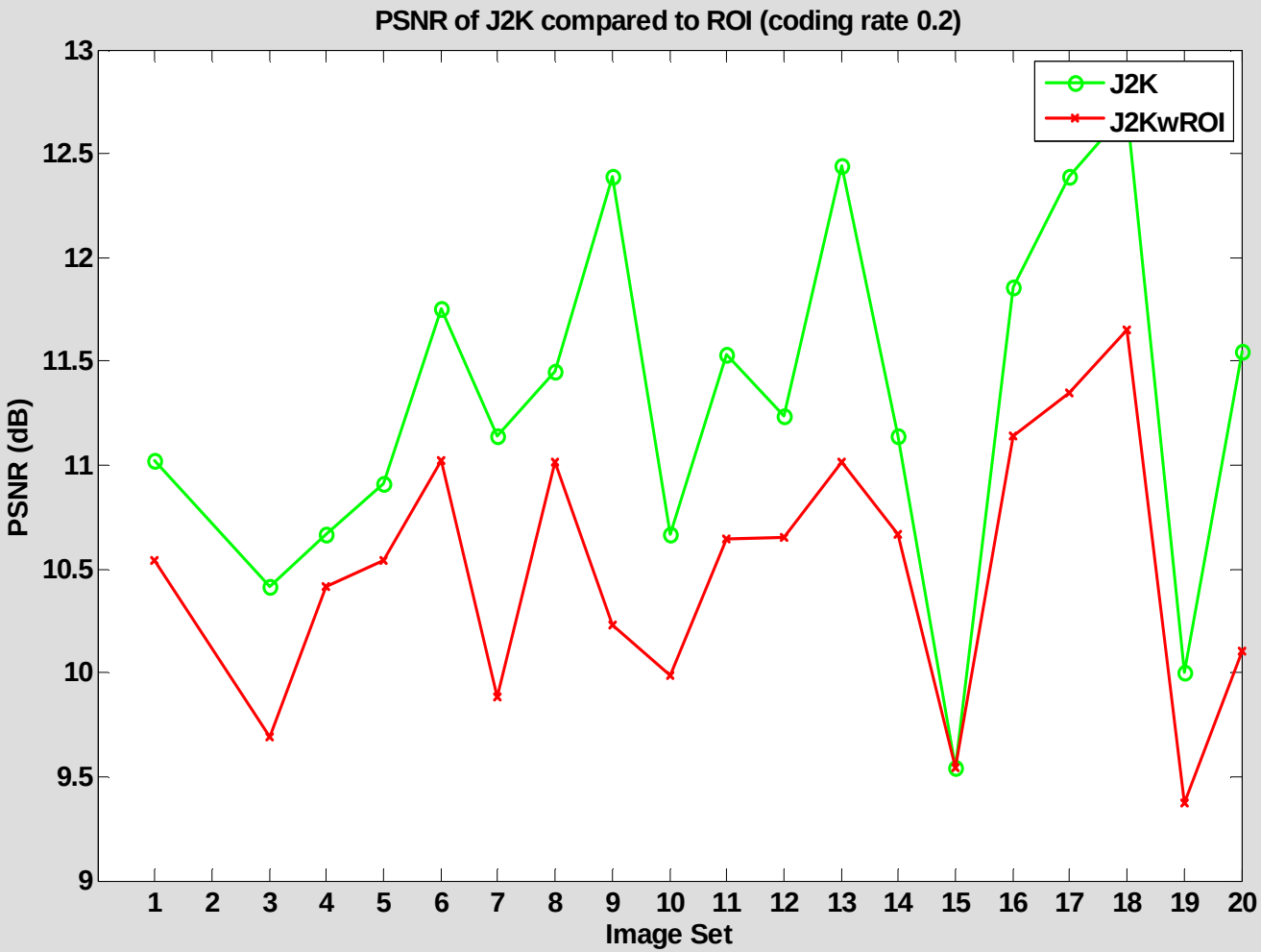
CASIA: False Negatives (False Reject)

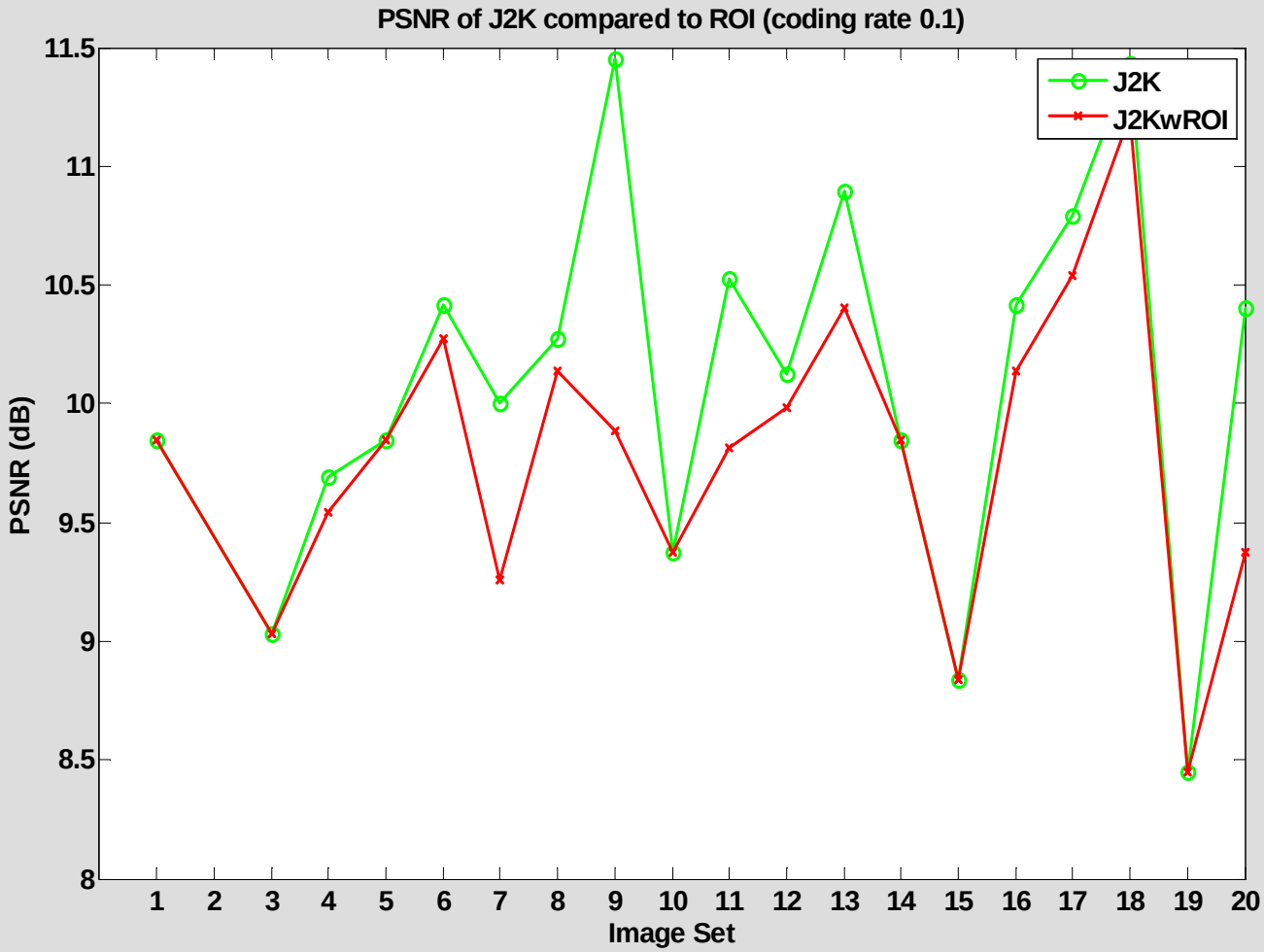


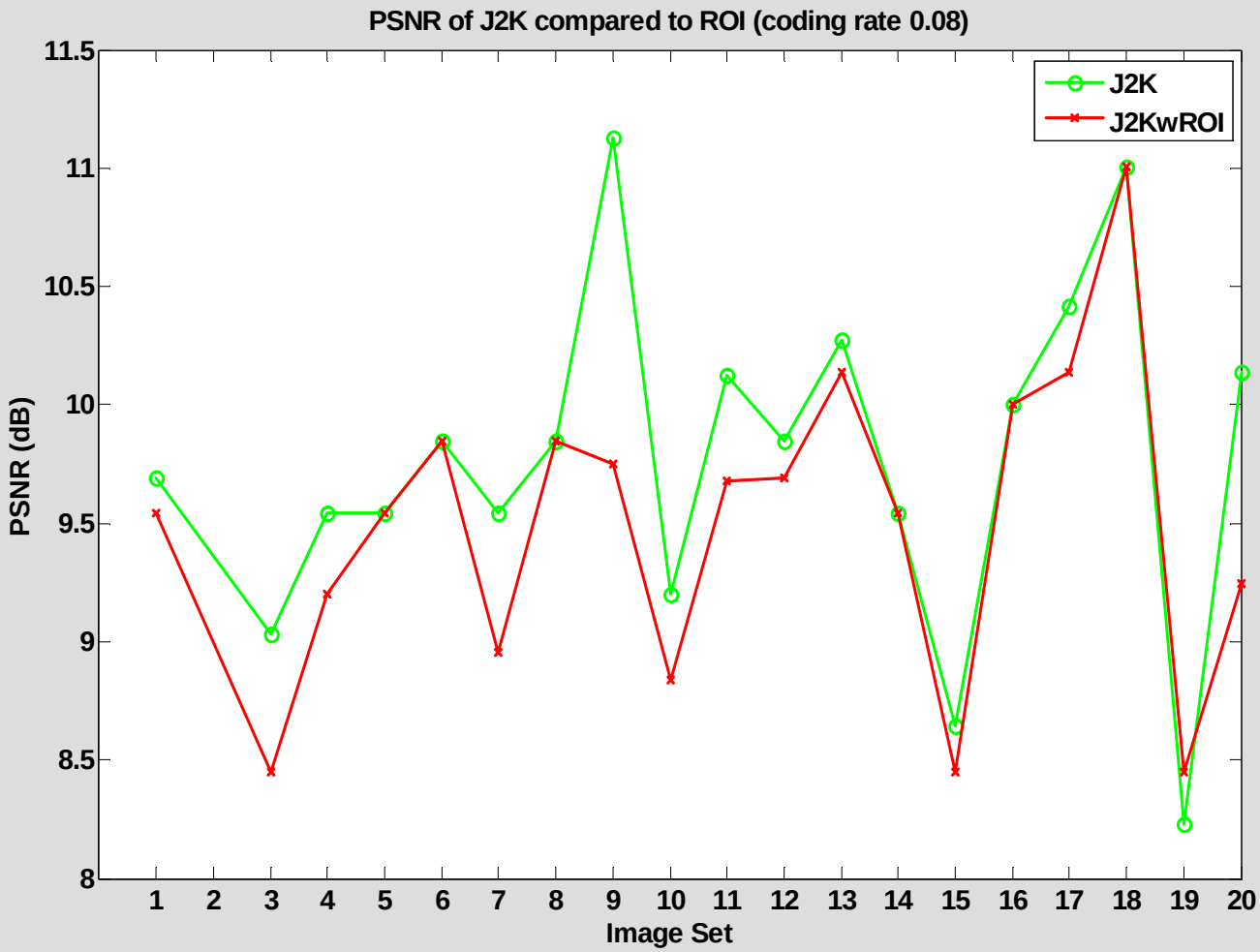
CASIA: Average Compressed File Size




PSNR of J2K compared to ROI (coding rate 0.3)
J2K
J2KwROI
PSNR (dB)
Image Set
14
13.5
13
12.5
12
11.5
11
10.5
10
9.5
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

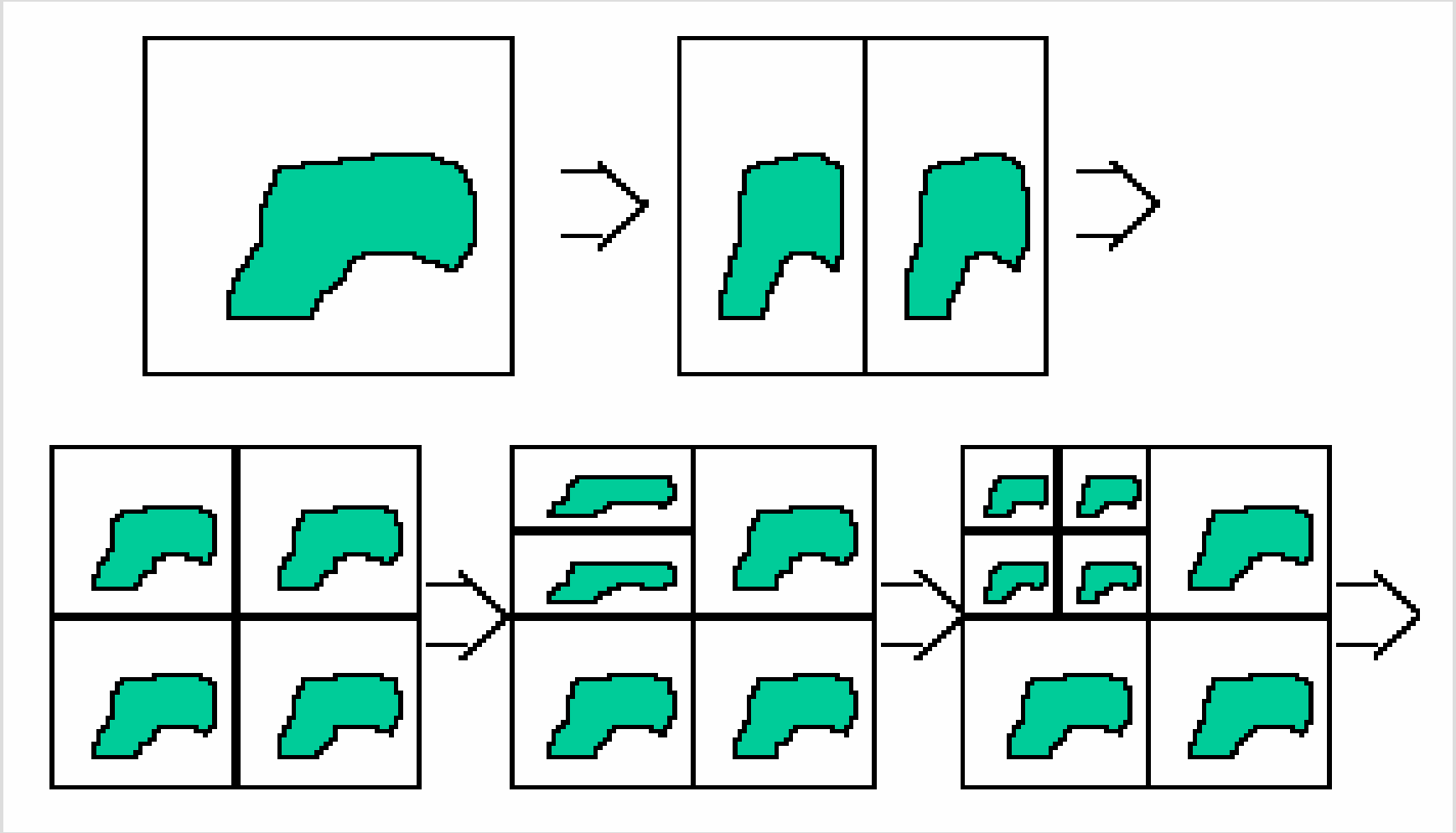

PSNR of J2K compared to ROI (coding rate 0.2)
J2K
J2KwROI
PSNR (dB)
Image Set
13
12.5
12
11.5
11
10.5
10
9.5
9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20


PSNR of J2K compared to ROI (coding rate 0.1)
J2K
J2KwROI
PSNR (dB)
Image Set
11.5
11
10.5
10
9.5
9
8.5
8
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20


PSNR of J2K compared to ROI (coding rate 0.08)
J2K
J2KwROI
PSNR (dB)
Image Set
11.5
11
10.5
10
9.5
9
8.5
8
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Danke für Ihre Aufmerksamkeit!

ROI Kodierung



MaxShift Methode