



# „Compressed domain image retrieval“

Christian Ott

*Seminar „Inhaltsbasierte Bildsuche“*

- Universität Freiburg -

04. Februar 2005

# Übersicht

1. Einleitung
2. JPEG
3. Merkmalsextraktion
4. Ausblick
5. Test-Ergebnisse
6. Matlab



# 1. Einleitung



Ähnliche Bilder



Merkmals-DB

Merkmals-Extraktion

# 1. Einleitung



Bilder sind oft in *komprimierter Form* gespeichert.

Merkmalsextraktion ist aufwendig bei bisherigen Verfahren (*vollständige Dekompression*).



**Bildsuche mit Merkmalen der komprimierten Bilder**

# 1. Einleitung

Es gibt viele Kompressionsverfahren für Bilder (gif, jpeg, jpeg2000, png, ...).

Unser Ansatz verwendet die Merkmale von komprimierten Bildern. Hierbei beschränken wir uns auf **jpeg**-Bilder.

# Übersicht

1. Einleitung
2. JPEG
3. Merkmalsextraktion
4. Ausblick
5. Test-Ergebnisse
6. Matlab



## 2. JPEG

*J.P.E.G. = Joint Photographic Expert Group*

**Das Verfahren basiert im wesentlichen auf 5 Schritten:**

1. Konvertierung in den YCbCr-Farbraum

2. Farb-Subsampling

 3. Diskrete Cosinus Transformation (DCT)

4. Quantisieren der DCT-Koeffizienten

5. Kodieren der Koeffizienten

## 2. JPEG

### 1. Konvertierung in den YCbCr-Farbraum:

YCbCr ist ein Helligkeit-Farbigkeit-Modell. Ein Farbwert wird durch eine Grundhelligkeit (**Y**) und dessen Abweichung von Grau in Richtung Blau (**Cb**) und Rot (**Cr**) bestimmt.

### 2. Farb-Subsampling:

Das Auge kann **feine Unterschiede nebeneinander liegender Werte** Cb bzw. Cr nur sehr schlecht wahrnehmen.

→ Farbwerte für Bereiche von 2x2 Pixeln werden gemittelt

## 2. JPEG

### 3. Diskrete Cosinus Transformation (DCT):

Eine Grafik wird mit **Blöcken (8x8 Pixel)** gerastert. Jeder Block wird mittels der DCT nach folgender Formel bearbeitet:

$$F(u, v) = \frac{1}{4} C(u)C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

$$\text{mit: } \left. \begin{array}{l} C(u), C(v) = \frac{1}{\sqrt{2}}, \text{ für } m, n = 0 \\ C(u), C(v) = 1, \text{ sonst} \end{array} \right\} \text{ Normierungskonstanten}$$

## 2. JPEG

### 3. Diskrete Cosinus Transformation (DCT):

Bei der DCT werden die 64 Pixelwerte in 64 Frequenzbereiche umgesetzt:

$$S_{jk} \quad \text{mit } j, k \in \{0, \dots, 7\}$$

$S_{00}$  wird als **DC-Koeffizient** (Gleichanteil) bezeichnet und entspricht dem **Frequenzanteil 0** in beiden Achsen. Er bestimmt den Grundfarbton.

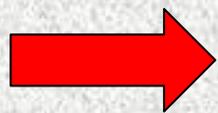
Die übrigen  $S_{jk}$  werden **AC-Koeffizienten** genannt. Diese repräsentieren mit steigendem Abstand zum DC-Wert höhere Frequenzen.

## 2. JPEG

### 3. Diskrete Cosinus Transformation (DCT):

Die DCT nutzt die Schwächen des menschlichen Auges und filtert die hohen Ortsfrequenzen heraus, die das Auge ohnehin nicht wahrnehmen kann.

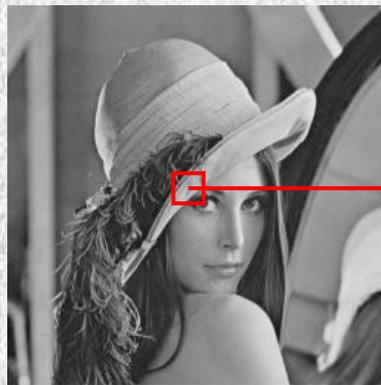
Da sich benachbarte Pixelwerte in der Regel kaum unterscheiden, werden nach der DCT nur der **DC-Koeffizient** und einige **niederfrequente AC-Koeffizienten** größere Werte annehmen.



Die anderen Koeffizienten werden fast Null sein bzw. meistens sogar gleich Null sein.

# 2. JPEG

## 3. Diskrete Cosinus Transformation (DCT):

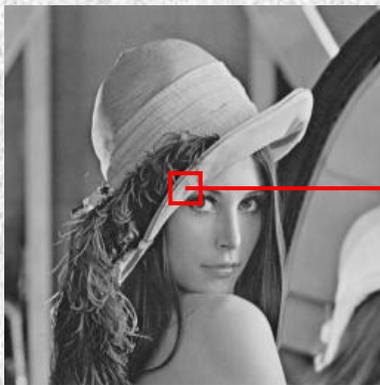


35	116	46	15	12	11	8	6	
19	86	75	14	12	9	11	55	
24	52	102	17	7	9	13	68	
57	66	141	49	6	10	52	127	
18	28	123	42	7	37	117	112	
8	9	80	81	25	109	117	84	
8	4	27	84	98	129	88	89	
6	5	14	67	110	83	83	140	

**8x8 Block Originalbild**

# 2. JPEG

## 3. Diskrete Cosinus Transformation (DCT):



$$\begin{bmatrix}
 415 & -75 & 17 & -110 & -50 & -3 & 36 & 11 \\
 -108 & 195 & 54 & -50 & -75 & -44 & -17 & 12 \\
 -41 & -2 & -49 & 95 & 35 & -61 & -53 & -35 \\
 8 & -7 & -16 & 31 & -51 & 41 & -18 & -23 \\
 22 & 7 & 47 & -37 & 17 & -10 & -13 & 20 \\
 20 & 15 & -6 & 14 & 18 & -11 & 10 & -18 \\
 -10 & -7 & 7 & -6 & -2 & 7 & 6 & -6 \\
 -17 & -14 & -4 & 9 & -2 & -5 & -3 & 12
 \end{bmatrix}$$

**8x8 Block nach DCT**

## 2. JPEG

### 4. Quantisieren der DCT-Koeffizienten:

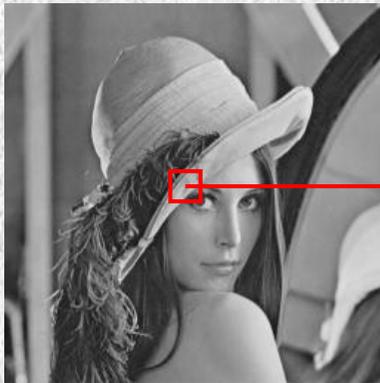
Die DCT-Koeffizienten werden durch einen **Quantisierungsfaktor geteilt** und auf den nächsten Integerwert **gerundet**.

Die Quantisierung erfolgt durch eine **Quantisierungsmatrix**. Jeder Koeffizient wird durch den entsprechenden Wert geteilt.

Durch die Rundungsfehler gehen Informationen verloren.

# 2. JPEG

## 4. Quantisieren der DCT-Koeffizienten:

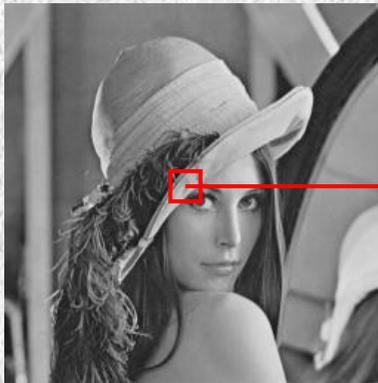


$$\begin{bmatrix}
 415 & -75 & 17 & -110 & -50 & -3 & 36 & 11 \\
 -108 & 195 & 54 & -50 & -75 & -44 & -17 & 12 \\
 -41 & -2 & -49 & 95 & 35 & -61 & -53 & -35 \\
 8 & -7 & -16 & 31 & -51 & 41 & -18 & -23 \\
 22 & 7 & 47 & -37 & 17 & -10 & -13 & 20 \\
 20 & 15 & -6 & 14 & 18 & -11 & 10 & -18 \\
 -10 & -7 & 7 & -6 & -2 & 7 & 6 & -6 \\
 -17 & -14 & -4 & 9 & -2 & -5 & -3 & 12
 \end{bmatrix}$$

**8x8 Block nach DCT**

# 2. JPEG

## 4. Quantisieren der DCT-Koeffizienten:

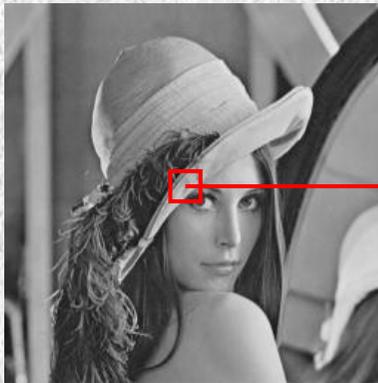


16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

**8x8 Block Quantisierungsmatrix**

# 2. JPEG

## 4. Quantisieren der DCT-Koeffizienten:



$$\begin{bmatrix}
 26 & -7 & 2 & -7 & -2 & 0 & 1 & 0 \\
 -9 & 16 & 4 & -3 & -2 & -1 & 0 & 0 \\
 -3 & 0 & -3 & 4 & 1 & -1 & -1 & -1 \\
 1 & 0 & -1 & 1 & -1 & 0 & 0 & 0 \\
 1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
 \end{bmatrix}$$

**8x8 Block nach Quantisierung**

## 2. JPEG

### 5. Kodieren der Koeffizienten:

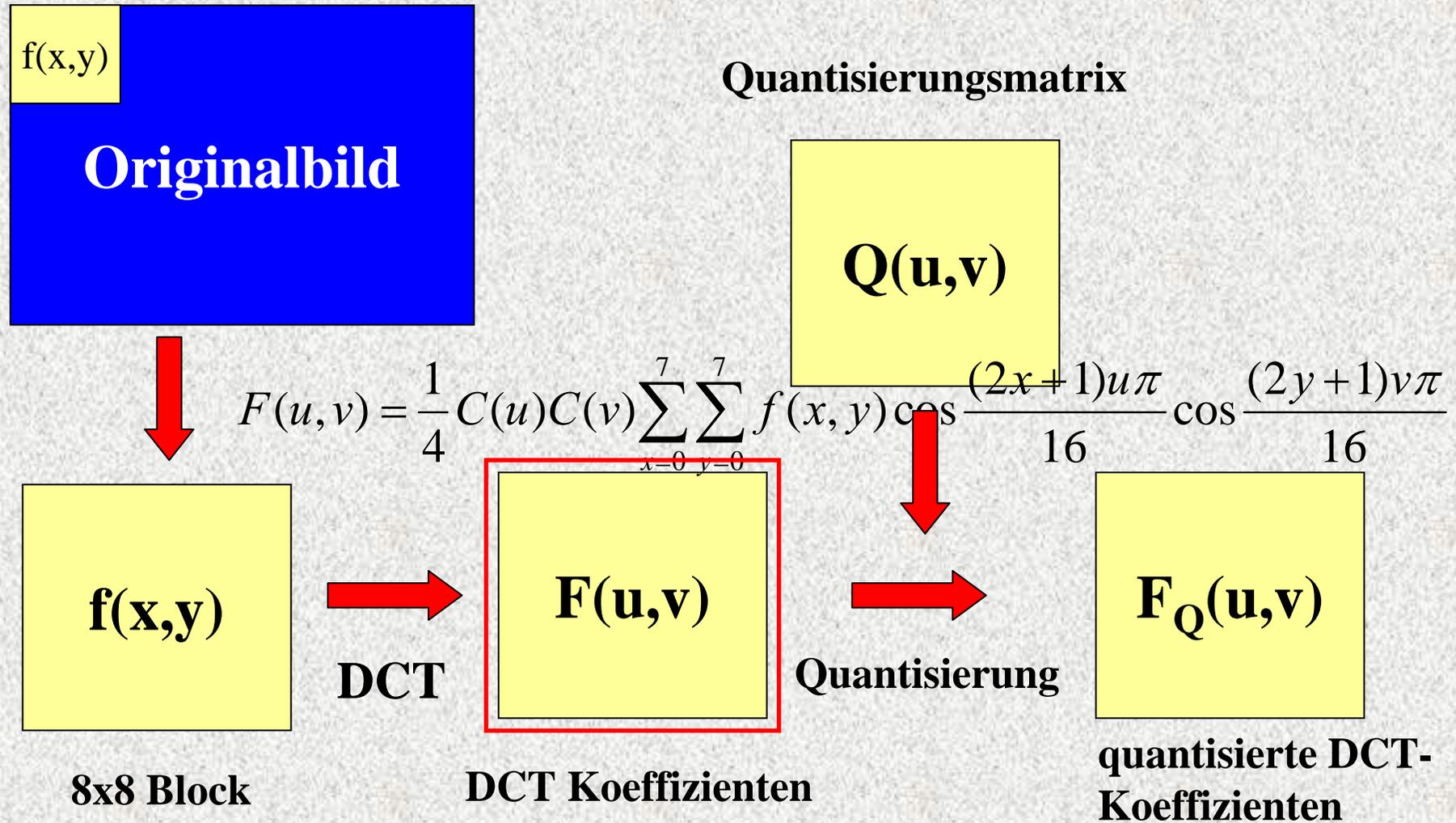
Die 64 Werte werden im **Zick-Zack** angeordnet.

Der erste Wert ist der DC-Koeffizient. Allerdings wird nur die Differenz zum DC-Koeffizienten im vorhergehenden Block kodiert.

Dadurch und durch die **Abarbeitung hin zu höheren Frequenzen** entstehen erneut kleinere Zahlen.

Um die Daten möglichst kompakt abzuspeichern, stellt der JPEG-Standard mehrere Verfahren bereit (z.B. **RLC, Huffman-Codierung**).

# Zusammenfassung JPEG:



# Übersicht

1. Einleitung
2. JPEG
3. Merkmalsextraktion
4. Ausblick
5. Test-Ergebnisse
6. Matlab



## 3. Merkmalsextraktion

Als Merkmal für die Bildsuche werden die **DCT-Koeffizienten** verwendet.

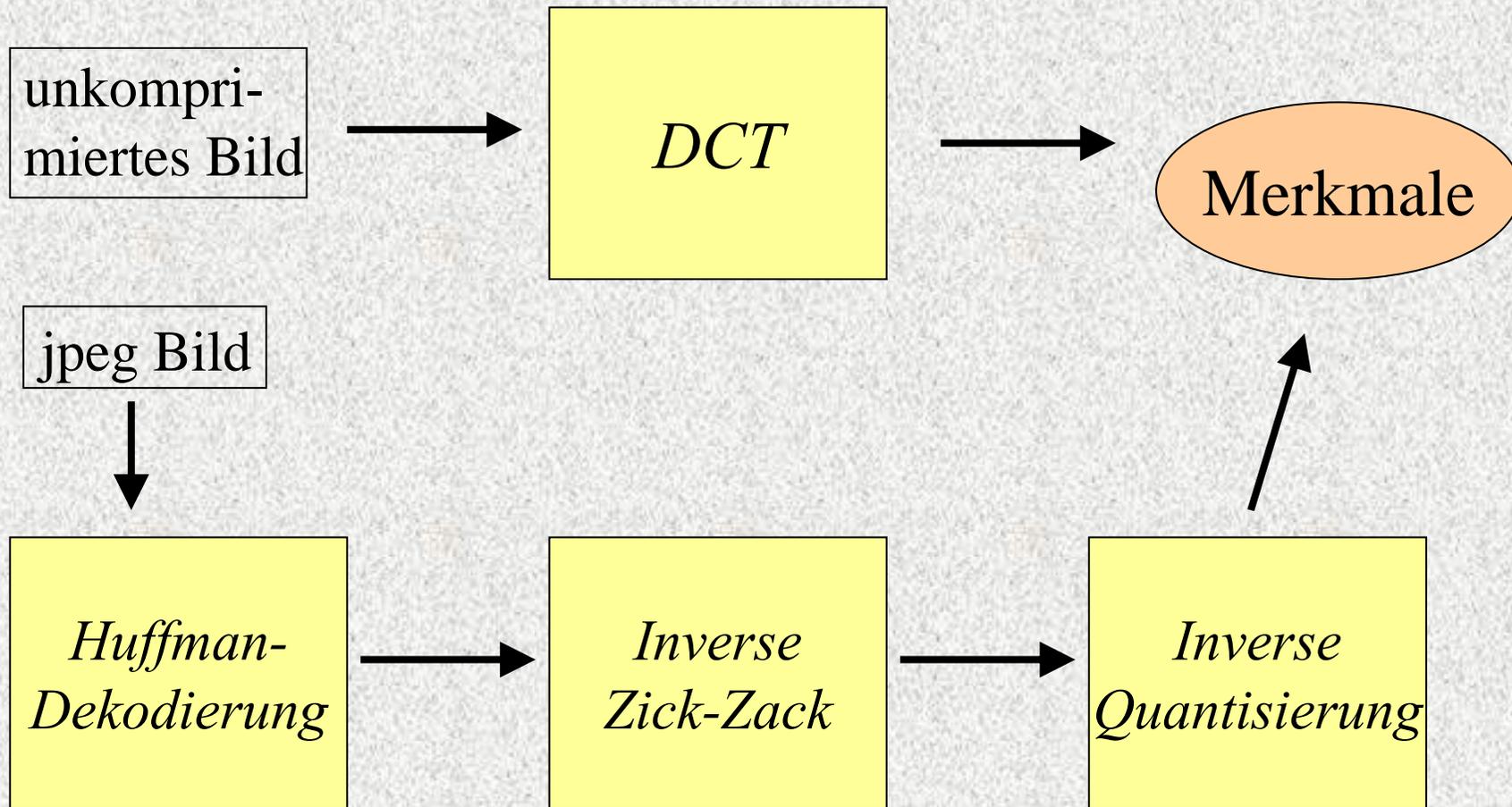
Bei **unkomprimierten Bildern** wird das Bild in 8x8 Blöcke aufgeteilt und blockweise die DCT-Koeffizienten berechnet.

Bei **jpeg-Bildern** wird eine *Huffman-Dekodierung* durchgeführt. Anschließend wird die *Zick-Zack*-Anordnung rückgängig gemacht und eine *inverse Quantisierung* angewandt, um die DCT-Koeffizienten zu erhalten.



**Keine vollständige Dekompression nötig. Der rechen-  
aufwendige Teil (*inverse DCT*) entfällt!**

### 3. Merkmalsextraktion



# 3. Merkmalsextraktion

Welche Information steckt in den DCT-Koeffizienten?



Originalbild  
 $M \times N$

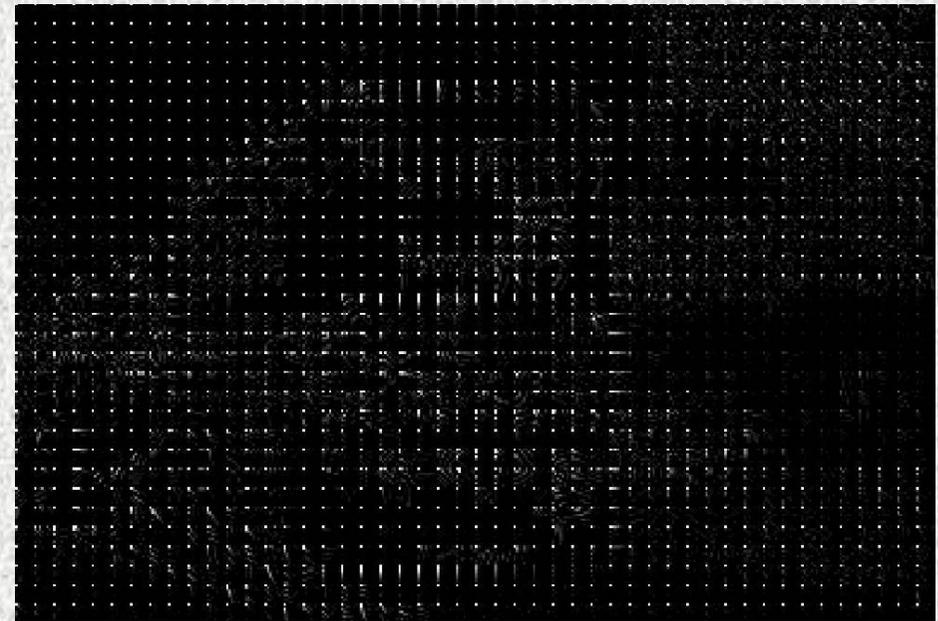


Bild nach DCT  
 $M \times N$

# 3. Merkmalsextraktion

Welche Information steckt in den DCT-Koeffizienten?



Originalbild  
 $M \times N$



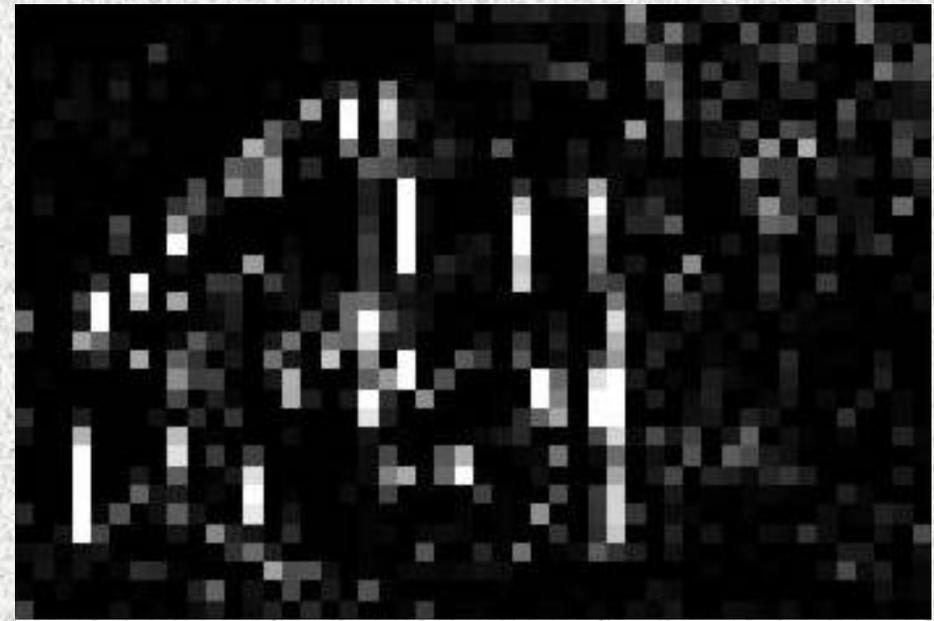
DC-Koeffizient  
 $M/8 \times N/8$

# 3. Merkmalsextraktion

Welche Information steckt in den DCT-Koeffizienten?



Originalbild  
 $M \times N$



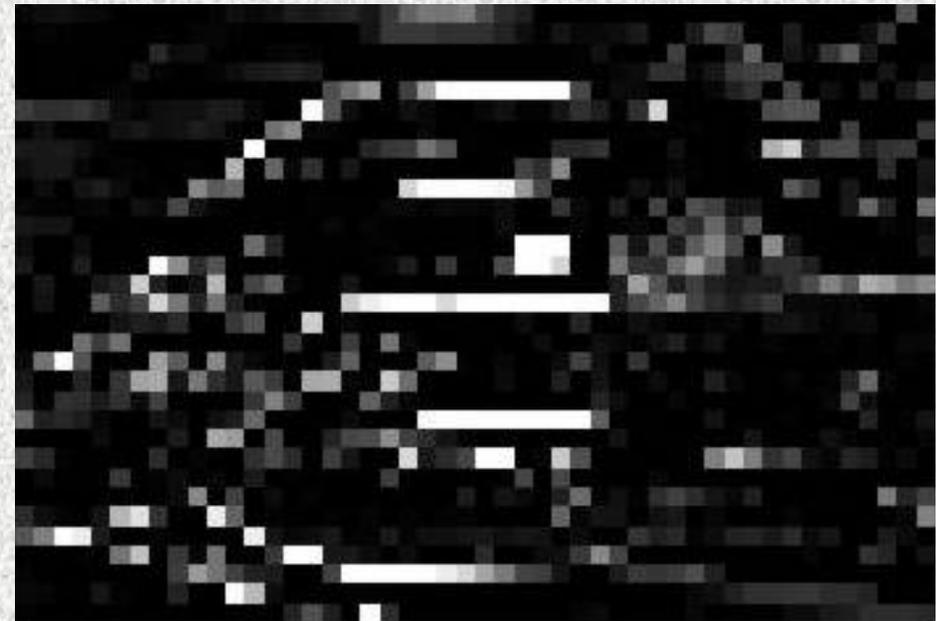
AC-1 Koeffizient  
 $M/8 \times N/8$

# 3. Merkmalsextraktion

Welche Information steckt in den DCT-Koeffizienten?



Originalbild  
 $M \times N$



AC-2 Koeffizient  
 $M/8 \times N/8$

### 3. Merkmalsextraktion

Berechnung der durchschnittlichen (absoluten) DCT-Koeffizienten:

Sei  $f(m,n)$  mit  $0 \leq m < M$ ,  $0 \leq n < N$  das Originalbild, wobei  $M \times N$  die Größe des Originalbildes ist.

Das Originalbild wird in  $8 \times 8$  große Teilblöcke unterteilt :

$$f_{kl}(i, j) \text{ mit } 0 \leq i < 8, 0 \leq j < 8, 0 \leq k < \frac{M}{8} = K, 0 \leq l < \frac{N}{8} = L,$$

wobei  $k$  und  $l$  den Teilblockindex angeben.

Die Anzahl der  $8 \times 8$  Teilblöcke beträgt  $KL$ .

### 3. Merkmalsextraktion

Berechnung der durchschnittlichen (absoluten) DCT-Koeffizienten:

$F_{kl}(u, v)$  bezeichne den DCT-Koeffizient des  $(k, l)$ -ten Teilblocks und  $u$  und  $v$  seien die Frequenzindizes.

Den durchschnittlichen (absoluten) DCT-Koeffizienten  $F^*(u, v)$  erhält man mit:

$$F^*(u, v) = \frac{1}{KL} \sum_k \sum_l |F_{kl}(u, v)|$$

Die Anzahl der DCT-Koeffizienten beträgt  $8 \times 8 = 64$ .

### 3. Merkmalsextraktion

Anstatt mit  $F^*(u,v)$  rechnen wir mit den logarithmierten Werten  $\log(F^*(u,v))$ .

Als Abstandsmaß definieren wir:

$$d(q, f) = \sum_{p=1}^{64} \frac{|Q(p) - F(p)|}{\sigma_Q}$$

$Q(p)$  ist der p.-te DCT-Koeffizient des Anfragebildes  $q(m,n)$

$F(p)$  ist der p.-te DCT-Koeffizient eines Bildes  $f(m,n)$  aus der DB.

$\sigma_Q$  ist die Standardabweichung von  $Q$ .

# 3. Merkmalsextraktion

Für die Bildsuche werden aber nicht alle der 64 Koeffizienten  $F^*(0,0), \dots, F^*(7,7)$  benötigt. Die Koeffizienten mit den hohen Frequenzanteilen können weggelassen werden.

$$F^* = \begin{bmatrix} F^*(0,0) & F^*(0,1) & F^*(0,2) & F^*(0,3) & F^*(0,4) & F^*(0,5) & F^*(0,6) & F^*(0,7) \\ F^*(1,0) & F^*(1,1) & F^*(1,2) & F^*(1,3) & F^*(1,4) & F^*(1,5) & F^*(1,6) & F^*(1,7) \\ F^*(2,0) & F^*(2,1) & F^*(2,2) & F^*(2,3) & F^*(2,4) & F^*(2,5) & F^*(2,6) & F^*(2,7) \\ F^*(3,0) & F^*(3,1) & F^*(3,2) & F^*(3,3) & F^*(3,4) & F^*(3,5) & F^*(3,6) & F^*(3,7) \\ F^*(4,0) & F^*(4,1) & F^*(4,2) & F^*(4,3) & F^*(4,4) & F^*(4,5) & F^*(4,6) & F^*(4,7) \\ F^*(5,0) & F^*(5,1) & F^*(5,2) & F^*(5,3) & F^*(5,4) & F^*(5,5) & F^*(5,6) & F^*(5,7) \\ F^*(6,0) & F^*(6,1) & F^*(6,2) & F^*(6,3) & F^*(6,4) & F^*(6,5) & F^*(6,6) & F^*(6,7) \\ F^*(7,0) & F^*(7,1) & F^*(7,2) & F^*(7,0) & F^*(7,4) & F^*(7,5) & F^*(7,6) & F^*(7,7) \end{bmatrix}$$

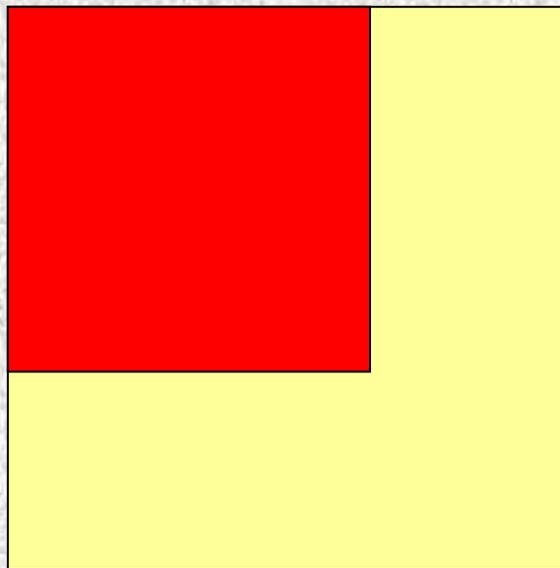
### 3. Merkmalsextraktion

Für die Bildsuche werden aber nicht alle der 64 Koeffizienten  $F^*(0,0), \dots, F^*(7,7)$  benötigt. Die Koeffizienten mit den hohen Frequenzanteilen können weggelassen werden.

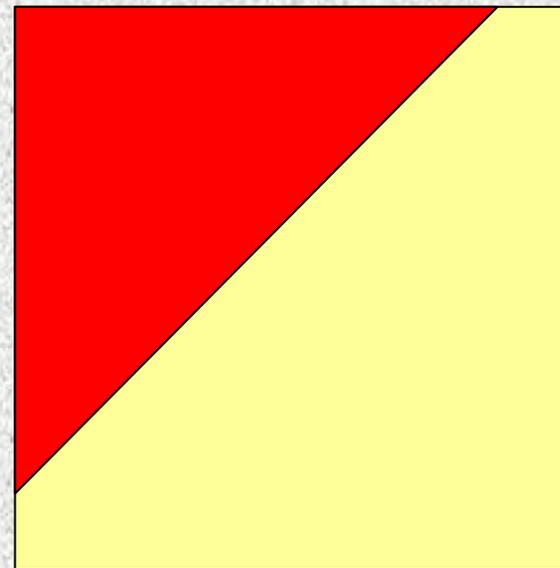
$$F^* = \begin{bmatrix} F^*(0,0) & F^*(0,1) & F^*(0,2) & F^*(0,3) & F^*(0,4) & F^*(0,5) & F^*(0,6) & F^*(0,7) \\ F^*(1,0) & F^*(1,1) & F^*(1,2) & F^*(1,3) & F^*(1,4) & F^*(1,5) & F^*(1,6) & F^*(1,7) \\ F^*(2,0) & F^*(2,1) & F^*(2,2) & F^*(2,3) & F^*(2,4) & F^*(2,5) & F^*(2,6) & 0 \\ F^*(3,0) & F^*(3,1) & F^*(3,2) & F^*(3,3) & F^*(3,4) & F^*(3,5) & 0 & 0 \\ F^*(4,0) & F^*(4,1) & F^*(4,2) & F^*(4,3) & F^*(4,4) & 0 & 0 & 0 \\ F^*(5,0) & F^*(5,1) & F^*(5,2) & F^*(5,3) & 0 & 0 & 0 & 0 \\ F^*(6,0) & F^*(6,1) & F^*(6,2) & 0 & 0 & 0 & 0 & 0 \\ F^*(7,0) & F^*(7,1) & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# 3. Merkmalsextraktion

Welche Koeffizienten sollen verwendet werden?



8 x 8



8 x 8

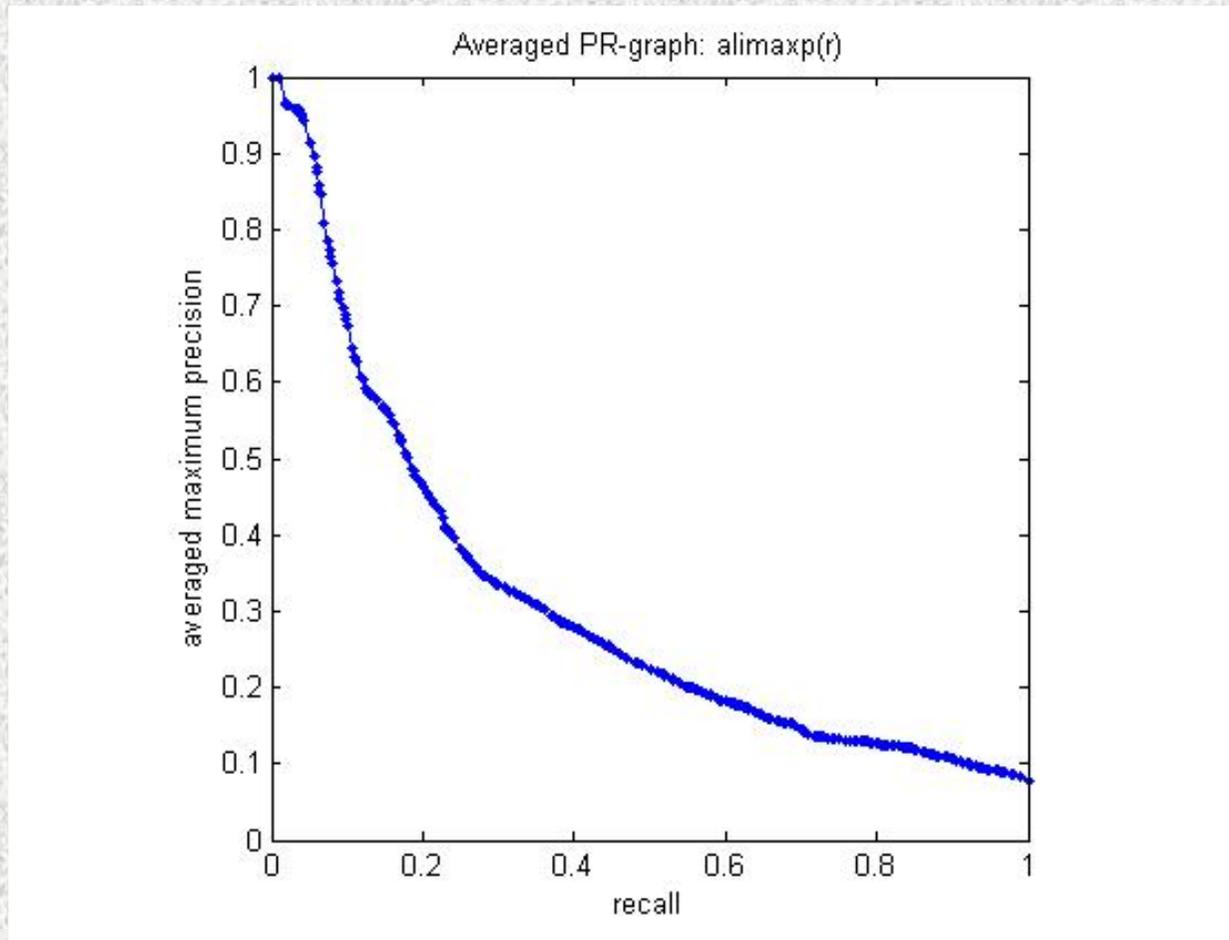
# 3. Merkmalsextraktion

$$F^* = \begin{bmatrix} F^*(0,0) & F^*(0,1) & F^*(0,2) & F^*(0,3) & 0 & 0 & 0 & 0 \\ F^*(1,0) & F^*(1,1) & F^*(1,2) & F^*(1,3) & 0 & 0 & 0 & 0 \\ F^*(2,0) & F^*(2,1) & F^*(2,2) & F^*(2,3) & 0 & 0 & 0 & 0 \\ F^*(3,0) & F^*(3,1) & F^*(3,2) & F^*(3,3) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$F^* = \begin{bmatrix} F^*(0,0) & F^*(0,1) & F^*(0,2) & F^*(0,3) & F^*(0,4) & 0 & 0 & 0 \\ F^*(1,0) & F^*(1,1) & F^*(1,2) & F^*(1,3) & 0 & 0 & 0 & 0 \\ F^*(2,0) & F^*(2,1) & F^*(2,2) & 0 & 0 & 0 & 0 & 0 \\ F^*(3,0) & F^*(3,1) & 0 & 0 & 0 & 0 & 0 & 0 \\ F^*(4,0) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

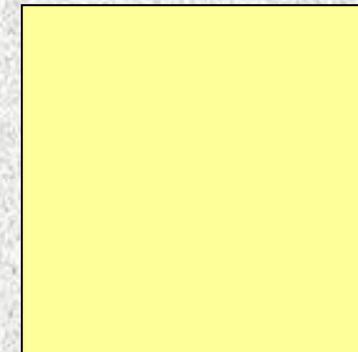
# 3. Merkmalsextraktion

Gemittelter PR-Graph für alle 17 Bilder:



Alle 64 DCT-Koeffizienten:

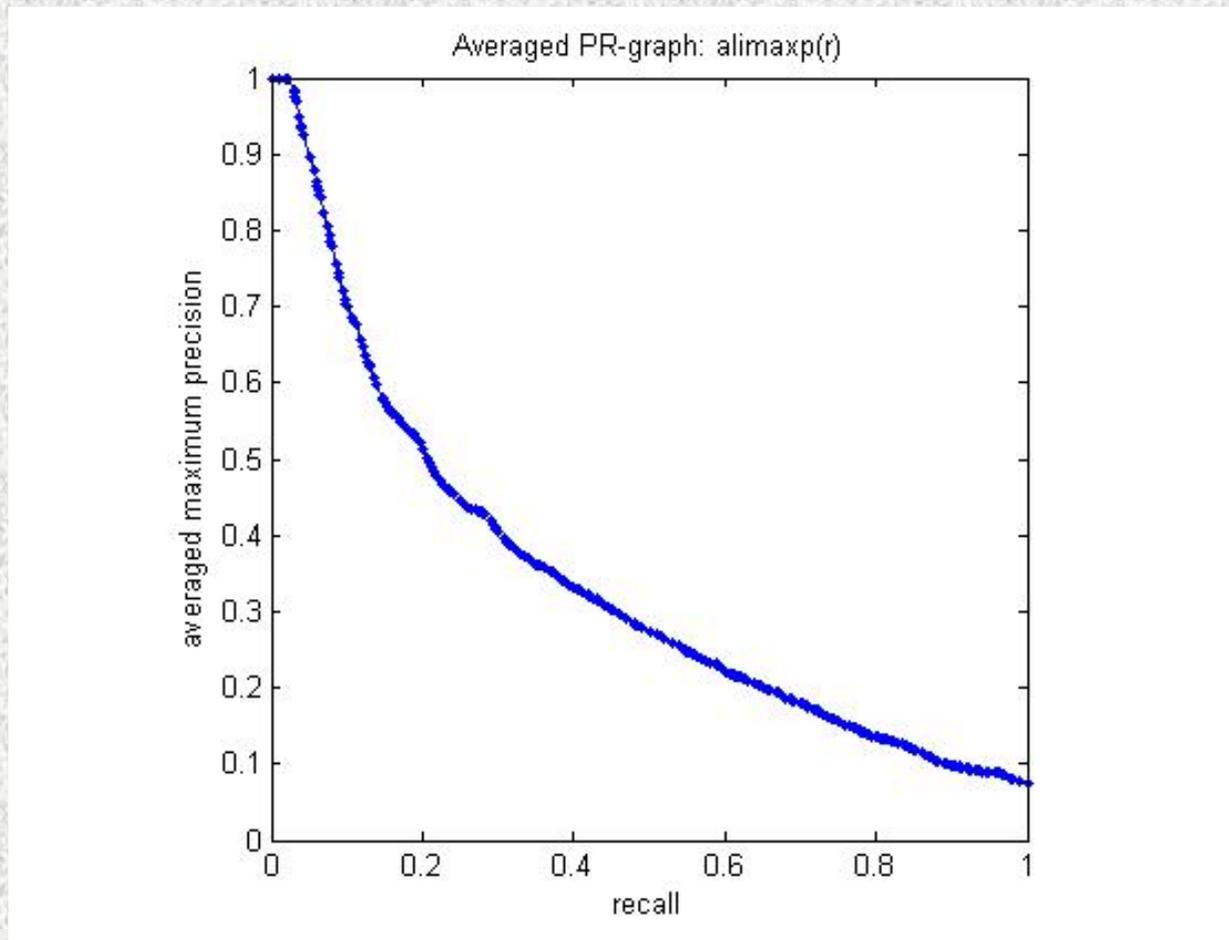
narank: 0.2037  
precision\_NR: 0.2641  
pr\_area: 0.2750



8 x 8

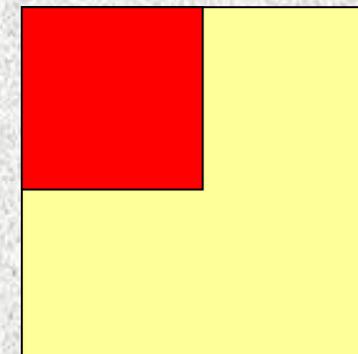
# 3. Merkmalsextraktion

Gemittelter PR-Graph für alle 17 Bilder:



16 DCT-Koeffizienten:

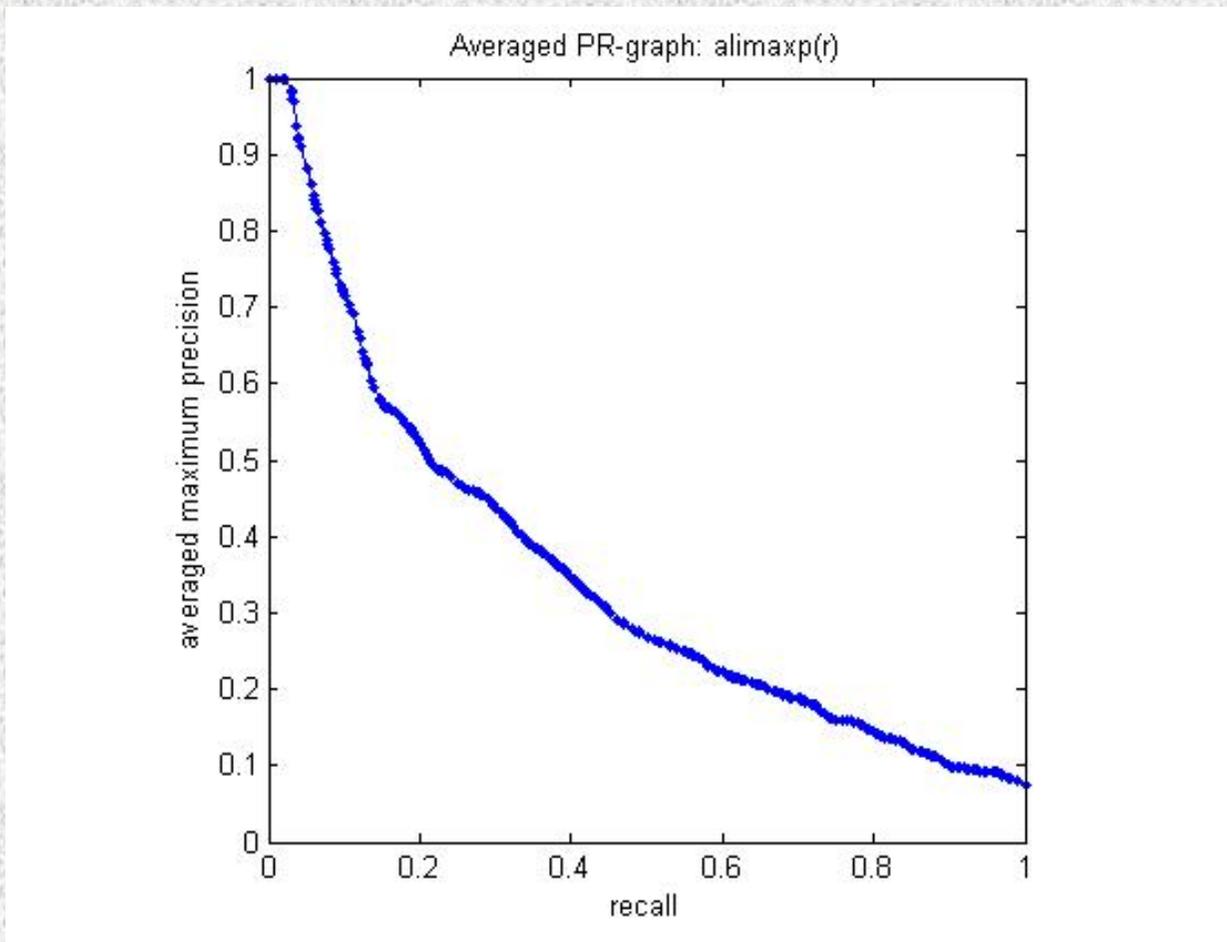
narank: 0.2000  
 precision\_NR: 0.3201  
 pr\_area: 0.3054



8 x 8

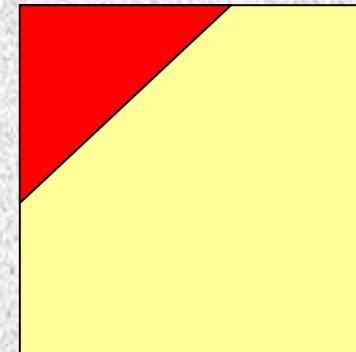
# 3. Merkmalsextraktion

Gemittelter PR-Graph für alle 17 Bilder:



15 DCT-Koeffizienten:

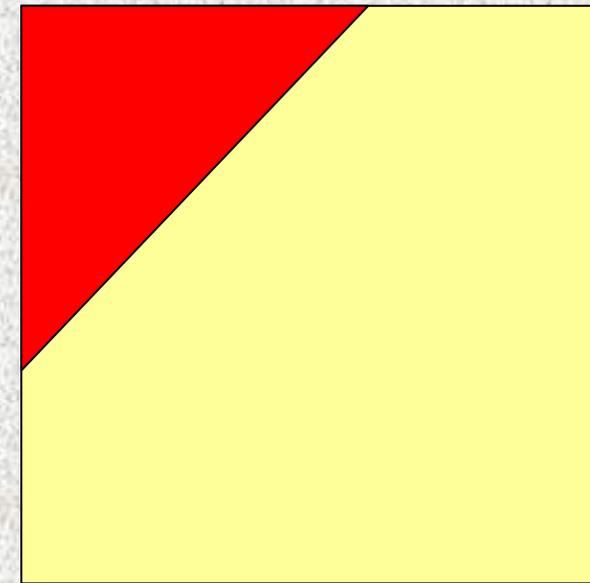
narank: 0.1898  
precision\_NR: 0.3305  
pr\_area: 0.3122



8 x 8

# 3. Merkmalsextraktion

Wir verwenden die 15 DCT-Koeffizienten im oberen Dreieck.



8 x 8

*Weitere Verbesserungsmöglichkeiten?*

# Übersicht

1. Einleitung
2. JPEG
3. Merkmalsextraktion
4. **Ausblick**
5. Test-Ergebnisse
6. Matlab



## 4. Ausblick

- DCT-Koeffizienten abhängig vom Anfragebild auswählen
- Farbinformation verwenden und in die Berechnung einbeziehen (zusätzliche Gewichtung Farbe/Grauwert)
- Klassifizierung der Bilder anhand der DCT-Koeffizienten (DC,  $AC_1$ ,  $AC_2$ ,  $AC_3$ )
- Region of interest
- ...

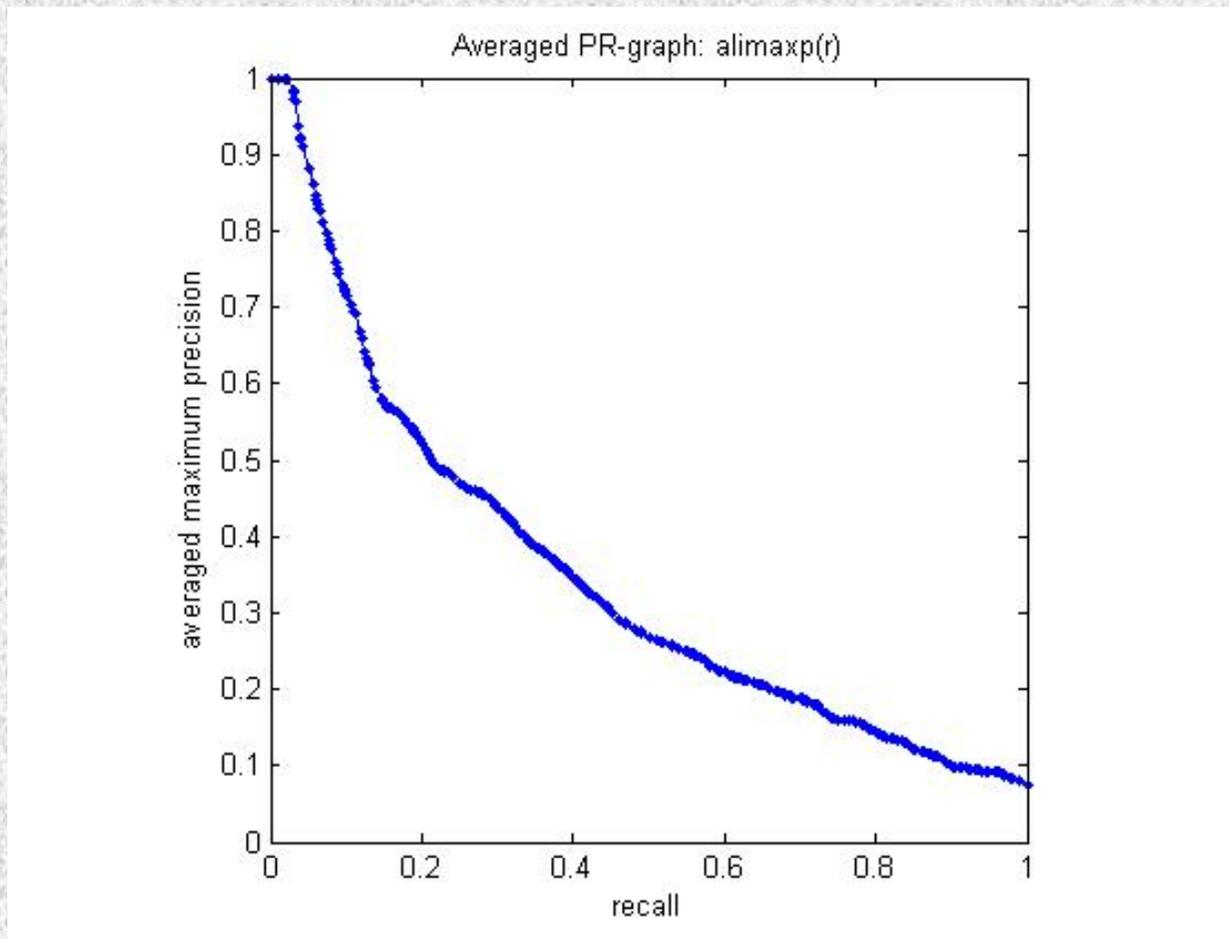
# Übersicht

1. Einleitung
2. JPEG
3. Merkmalsextraktion
4. Ausblick
5. Test-Ergebnisse
6. Matlab



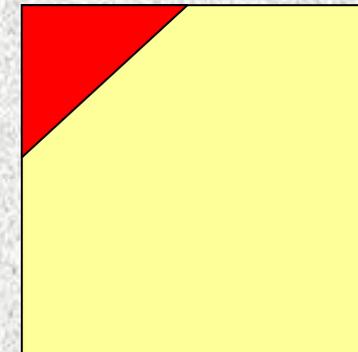
# 5. Test-Ergebnisse

Gemittelter PR-Graph für alle 17 Bilder:



15 DCT-Koeffizienten:

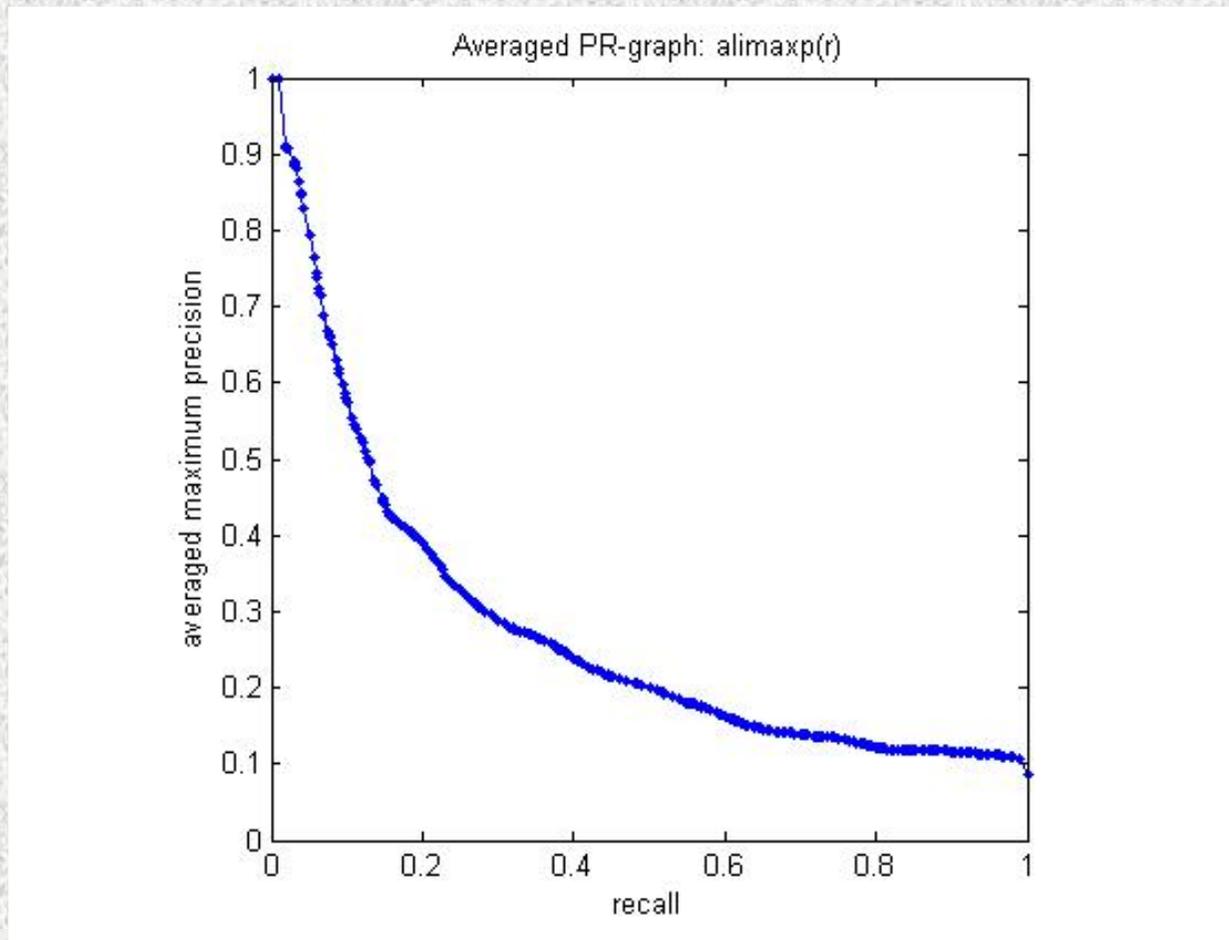
narank: 0.1898  
precision\_NR: 0.3305  
pr\_area: 0.3122



8 x 8

# 5. Test-Ergebnisse

Gemittelter PR-Graph für alle 17 Bilder:

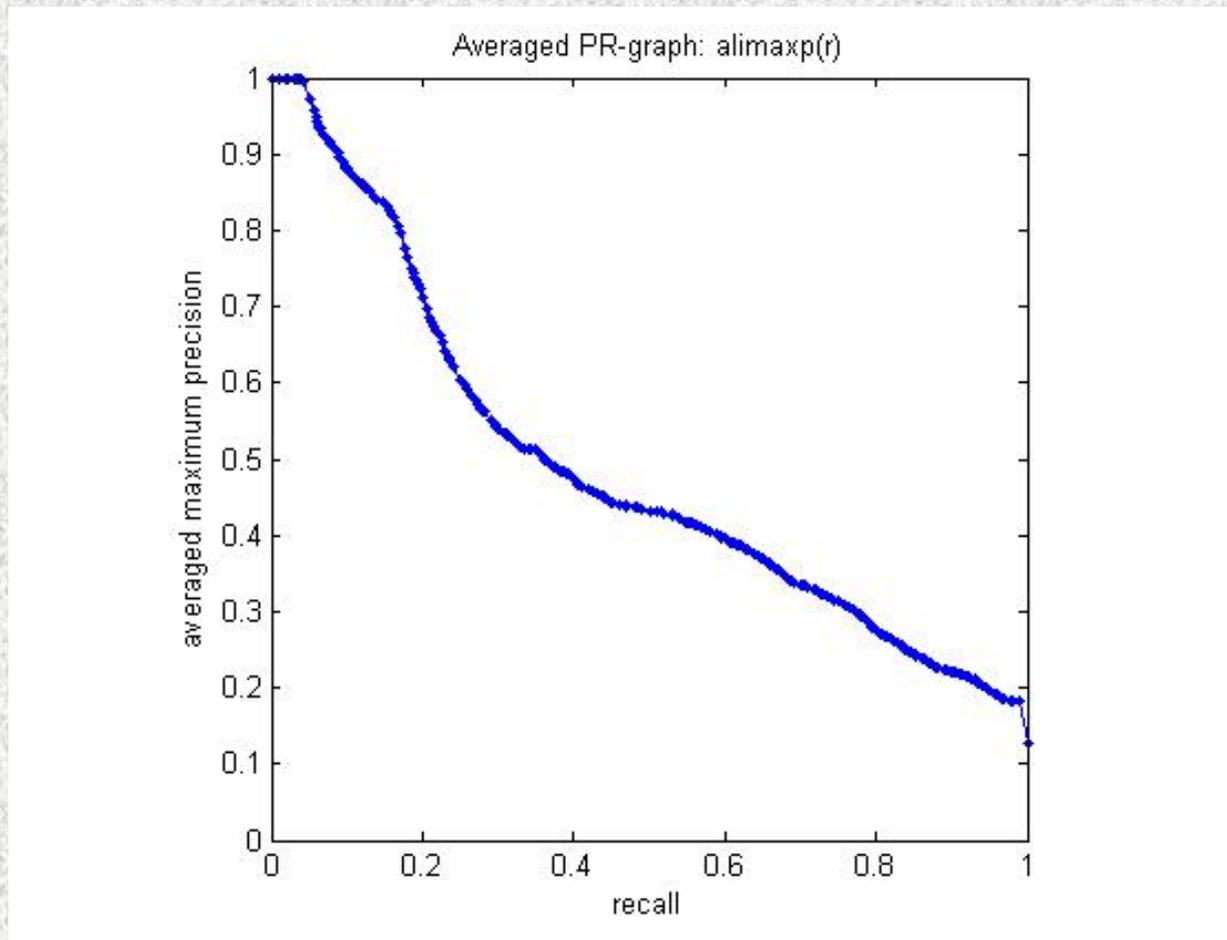


**SIMBA, Grauwert**

narank: 0.2391  
precision\_NR: 0.2472  
pr\_area: 0.2411

# 5. Test-Ergebnisse

Gemittelter PR-Graph für alle 17 Bilder:



SIMBA mit Farbe

narank: 0.1080  
precision\_NR: 0.4352  
pr\_area: 0.4593

61.jpg



2.jpg



87.jpg



47.jpg



934.jpg



51.jpg



5.jpg



262.jpg



903.jpg



64.jpg



82.jpg



708.jpg



900.jpg



166.jpg



860.jpg



119.jpg



176.jpg



813.jpg



904.jpg



597.jpg



115.jpg



830.jpg



290.jpg



154.jpg



167.jpg



506.jpg



283.jpg



239.jpg



248.jpg



527.jpg



298.jpg



299.jpg



574.jpg



278.jpg



280.jpg



227.jpg



201.jpg



331.jpg



578.jpg



300.jpg



313.jpg



392.jpg



324.jpg



388.jpg



341.jpg



359.jpg



358.jpg



326.jpg



371.jpg



328.jpg



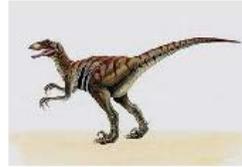
259.jpg



362.jpg



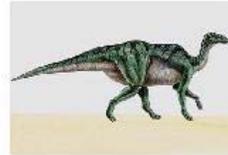
400.jpg



444.jpg



487.jpg



479.jpg



483.jpg



404.jpg



436.jpg



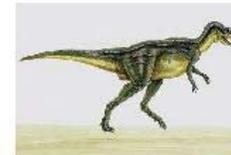
466.jpg



471.jpg



441.jpg



465.jpg



442.jpg



411.jpg



592.jpg



584.jpg



595.jpg



544.jpg



529.jpg



589.jpg



73.jpg



597.jpg



593.jpg



101.jpg



115.jpg



562.jpg



961.jpg



641.jpg



691.jpg



623.jpg



610.jpg



627.jpg



600.jpg



611.jpg



615.jpg



631.jpg



647.jpg



621.jpg



602.jpg



696.jpg



701.jpg



706.jpg



725.jpg



223.jpg



575.jpg



745.jpg



786.jpg



756.jpg



769.jpg



588.jpg



515.jpg



729.jpg



758.jpg



# Übersicht

1. Einleitung
2. JPEG
3. Merkmalsextraktion
4. Ausblick
5. Test-Ergebnisse
6. Matlab





Vielen Dank für die Aufmerksamkeit!

Anregungen, Fragen ???





## [Literaturliste]

[1] D. G. Sin, H. K. Kin, and R. H. Park , “Fast texture description and retrieval of DCT based compressed images”, *Electronics Letters*, vol. 37, no. 1, pp. 18-19, 2001.

[2] M. Shneier and M. Abdel-Mottaleb, “Exploiting the JPEG compression scheme for image retrieval”, *IEEE Trans. on Pattern Analysis & Mach. Intell*, vol. 18, No. 8, pp. 849-853, August 1996.

[3] R. Chang, W. Kuo, and H. Tsai, “Image Retrieval on Uncompressed and Compressed Domains”, *IEEE ICIP*, 2000.

## [Online-Quellen]

[1] <http://e-technik.naurod.com/studium/projekt/>