

Universität Kaiserslautern

Multimediale Informationssysteme (MMIS)
bei der Arbeitsgruppe Datenbanken und
Informationssysteme

5. Juli 2002

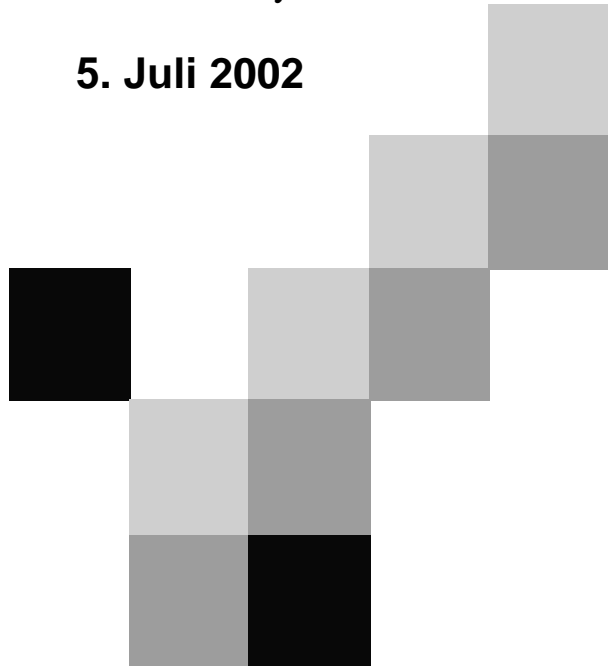


Image Retrieval

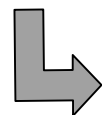
Was ist Image Retrieval ?

- Die Suche nach Bildern mit bestimmten Merkmalen

Früher: Nur wenige Bilder in digitaler Form

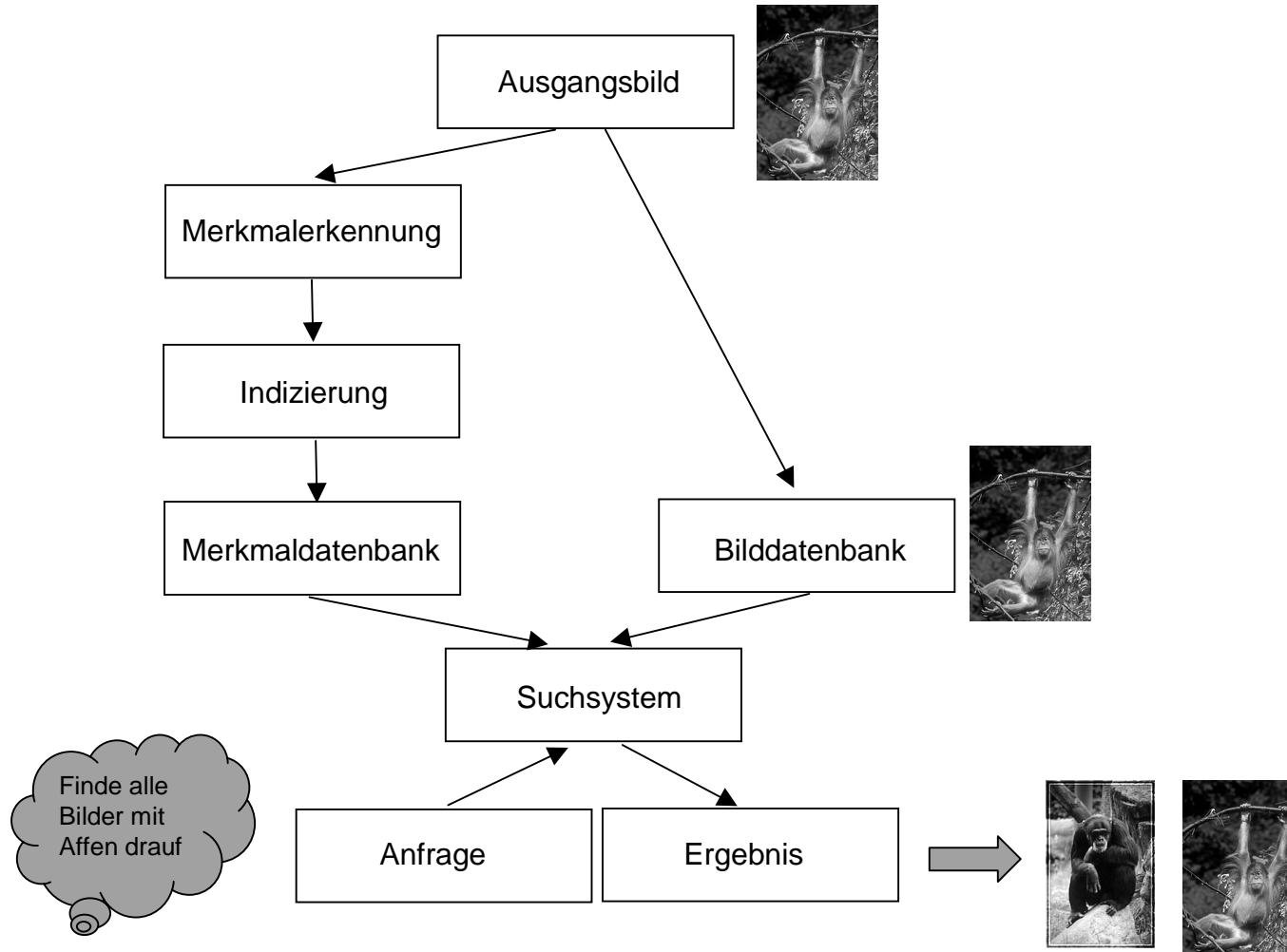
Heute: Immer mehr Bilder werden
in digitaler Form gespeichert

Geringe Kosten
für Speicher



Suche nach Algorithmen und Methoden,
die eine effiziente und genaue Suche ermöglichen

Überblick zur Bildsuche



Besonderheiten bei der Suche nach Bildern

Szenario:

- Polizist findet Photo
- Er möchte nun seine Verbrecherkartei durchsuchen
- Finde alle **Bilder**, die **ähnlich** zu diesem hier sind und gib mir die Namen der Personen zurück, die darauf zu sehen sind“.

Wesentliche Unterschiede zu bisherigen (textuellen) Anfragen:

1. Die Anfrage enthält ein Bild.
2. Die Anfrage ist ungenau, was hier durch das Wort *ähnlich* dargestellt werden soll, was dazu führt, dass man eine neue, „ungenaue“ Art der Suche benötigt.

Unterteilung eines Bildes:

Pixel:

- RGB-Anteil
- RGB Modell verwendet Grundfarben zur additiven Farbmischung

Zelle:

Besteht aus Tripel (*Name, Werte, Methode*)

- *Name* beschreibt die Eigenschaft
- *Werte* zulässige Werte der Eigenschaft
- *Methode*, die die Eigenschaft verarbeitet

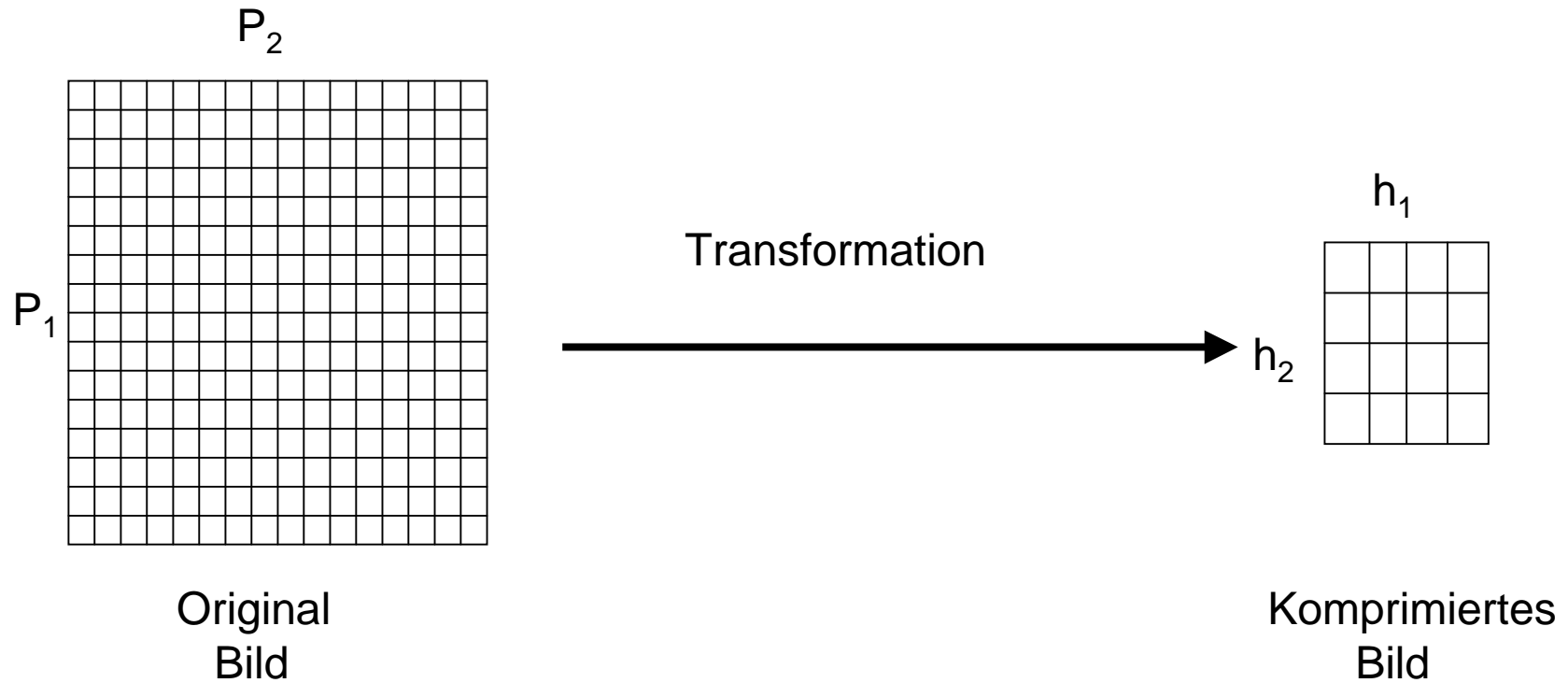
- Bei einem Bild, das Graustufen als Eigenschaft hat, könnte eine Zelle folgendermaßen dargestellt werden:

(graustufe, [0,1], graualgo)



Immer noch zu viele Informationen

Bildkomprimierung



Es passt nur ein einziges Bild auf eine Diskette

1. Transformationen zur Bildkomprimierung:

The Discrete Fourier Transformation (DFT)

- Grundlegende Idee: Jede periodische Funktion lässt sich als Summe von Sinus- und Cosinusfunktionen darstellen

Eigenschaften:

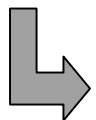
- Lange Rechenzeit
- Man kann das original Image aus der DFT Repräsentation wiedergewinnen, indem man die inverse Transformation DFT^{-1} auf $DFT(I)$ anwendet.
- DFT wird meist nicht alleine verwendet \Rightarrow Verlust der Invertierbarkeit

Transformationsbeispiele

2. Transformationen zur Bildkomprimierung:

The Discrete Cosine Transform (DCT)

1. Quellbild wurde in Blöcke von je 8x8 Punkten zerlegt
2. Jeder Block wird mittels DCT codiert (erster Wert des Blockes DC, restliche Werte AC)
3. Werte untereinander werden als Differenzen gespeichert
4. Frequenzen des DCT-Blocks vorzugsweise niedrig
5. Es ergibt sich somit:
 1. Ein DC Wert mit niedriger Frequenz
 2. Wenige niedrige AC Werte und viele Werte gleich null



Quantifizierung und Lauflängencodierung

Kodierung

Segmentierung

1. Erkennung von Merkmalen
2. Einteilung des Bildes in homogene Bereiche
3. Anwenden eines Homogenitätsprädikats
4. Worst-Case: Zelle entspricht einem Pixel

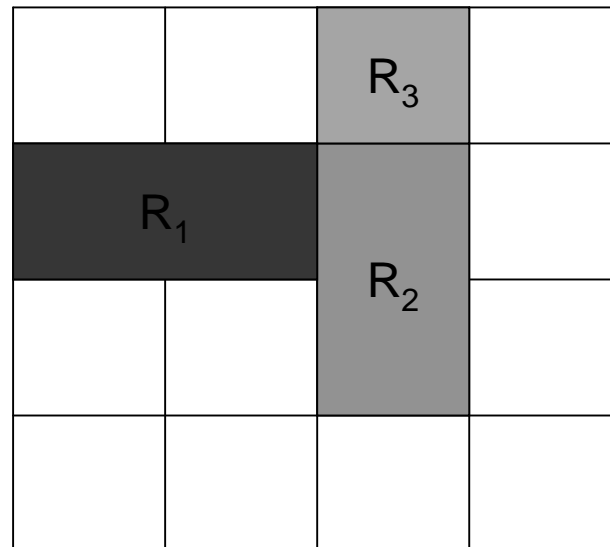


Bild mit 3 „*connected regions*“

Homogenitätsprädikate

Ein Homogenitätsprädikat in Bezug auf ein Bild I ist eine Funktion H die eine beliebige *connected region* \mathcal{R} als Eingabe nimmt und entscheidet, ob diese Region homogen ist.

- Sei δ eine reale Zahl mit $0 < \delta \leq 1$
- Gegeben sei ein schwarz-weiß Bild
- Ein einfaches Homogenitätsprädikat wäre zum Beispiel H :
- H liefert „true“ zurück, wenn über $(100 * \delta) \%$ der Zellen in Region R dieselbe Farbe haben

Anwendung eines Homogenitätsprädikat

Region	Zahl der schwarzen Pixel	Zahl der weißen Pixel
R ₁	800	200
R ₂	900	100
R ₃	100	900

Gegebene Regionen

Region	$H_{sw,0.8}$	$H_{sw,0.9}$	$H_{sw,0.95}$
R ₁	true	false	false
R ₂	true	true	false
R ₃	true	true	false

Ergebnis dreier Homogenitätsprädikate

Weitere Beispiele für Homogenitätsprädikate

Eine zweite Methode ist, dass man zuerst alle bw-levels in einer gewissen Spanne klassifiziert:

- bw-levels zwischen 0 und 0.1 die 1, zwischen 0.1 und 0.2 die 2 usw. bis zwischen 0.9 und 1 die 10 zugewiesen.
- δ ist wieder wie oben definiert.
- $H^{\text{class}}(R)$ liefert „true“ zurück, falls $(100 * \delta)$ % der Zellen aus der Region R in die selbe Klassifizierung fallen.

Eine dritte Methode ist eine dynamische Version von H^{class} :

- Das Prädikat $H_{\text{dyn}, \eta, \delta}$ liefert „true“ zurück, wenn $(100 * \delta)$ % der Zellen der Region R innerhalb eines Bereiches η einer realen Zahl r liegen, wobei r zur Laufzeit festgelegt wird.

Anwendung eines Homogenitätsprädikats

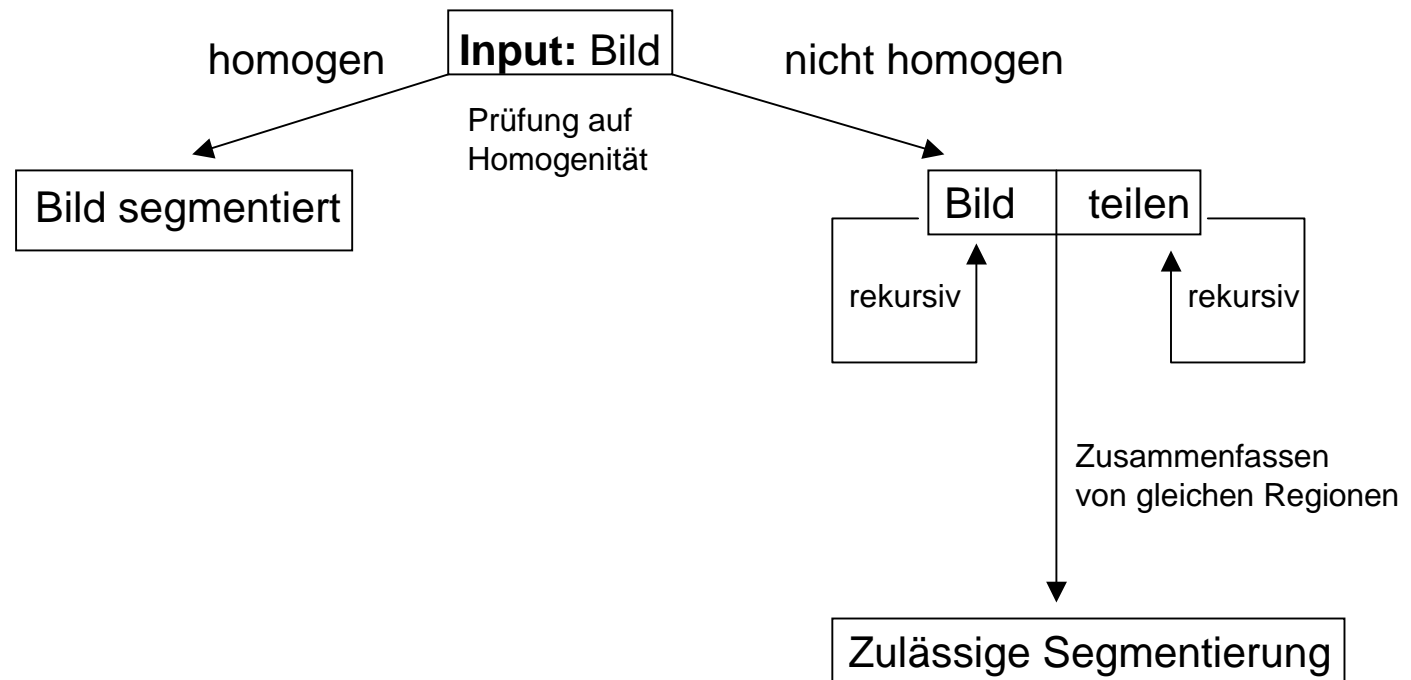
Zeile/Spalte	1	2	3	4
1	0.1	0.25	0.5	0.5
2	0.05	0.30	0.6	0.6
3	0.35	0.30	0.55	0.8
4	0.6	0.63	0.85	0.90

Wir wenden nun das Homogenitätsprädikat $H_{\text{dyn},0.05,1}$ an, das besagt, dass eine Region homogen ist, wenn ein r existiert, so dass jede Zelle in der Region ein bw-level v hat, so dass $|v - r| \leq 0.05$

Ergebnis des Homogenitätsprädikats

4	0.6	0.63	0.85	0.90
3	0.35	0.30	0.55	0.8
2	0.05	0.30	0.6	0.6
1	0.1	0.25	0.5	0.5
	1	2	3	4

Segmentierung mit Homogenitätsprädikaten



Ähnlichkeitsbasiertes Suchen

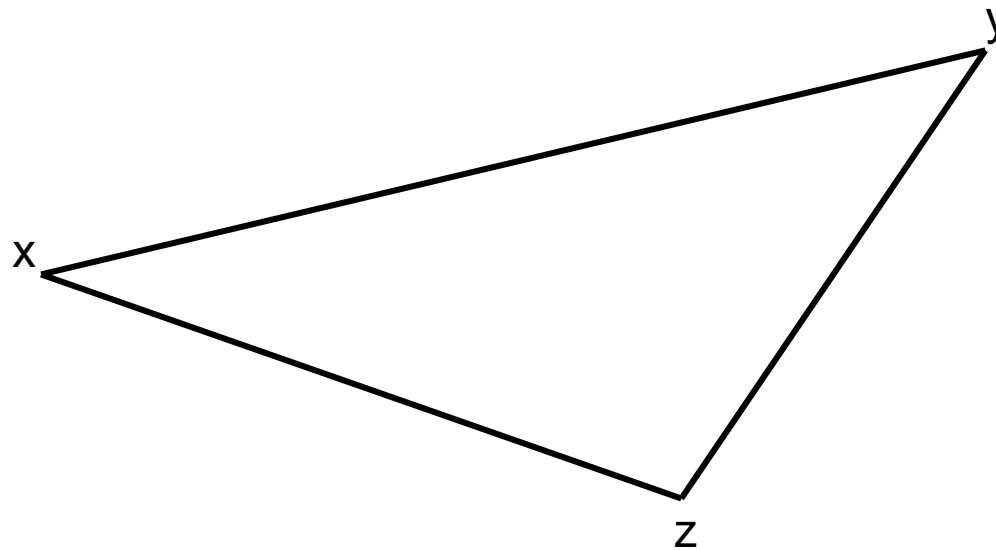


Was ist „**ÄHNLICH**“ ?

Metrische Näherungsverfahren

- Distanzmetrik d , die es ermöglicht, zwei Bilder miteinander zu vergleichen.
- Um so kürzer die Distanz zwischen zwei Bildern ist, um so ähnlicher sind sie sich.
- Bei dieser Art der Näherung würde eine Suche dann folgendermaßen aussehen:
 - „Gegeben sei ein Image I , finde den nächsten Nachbar von I im Bildarchiv.“
- Distanzfunktion muss folgende Eigenschaften erfüllen:
 - $d(x, y) = d(y, x)$ (Symmetrie)
 - $d(x, y) \leq d(x, z) + d(z, y)$ (Transitivität)
 - $d(x, x) = 0$ (Reflexivität)

Metrische Näherungsverfahren



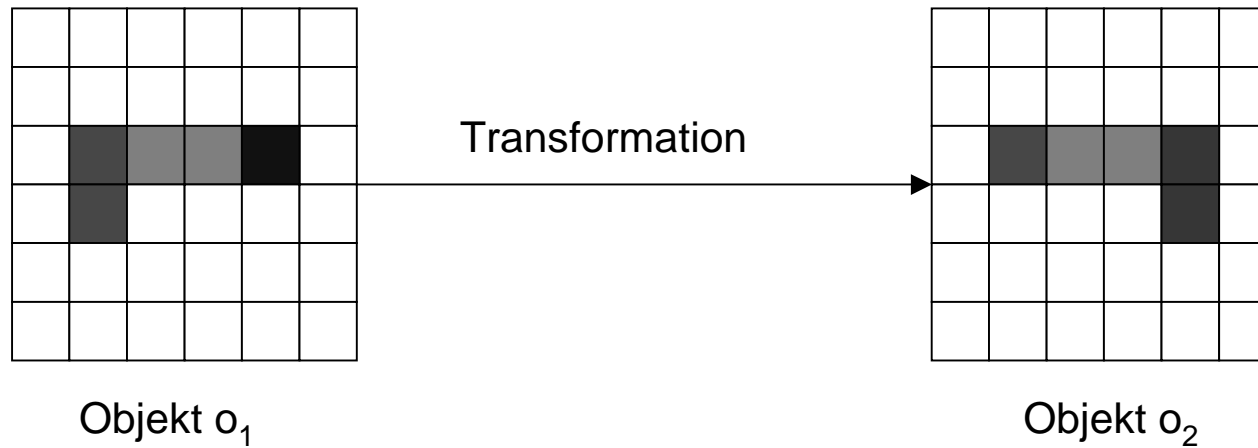
1. Distanz von x nach y genauso lang wie von y nach x
2. Kürzeste Distanz von x nach y
3. Distanz von x zu sich selber 0

Transformationsnäherung

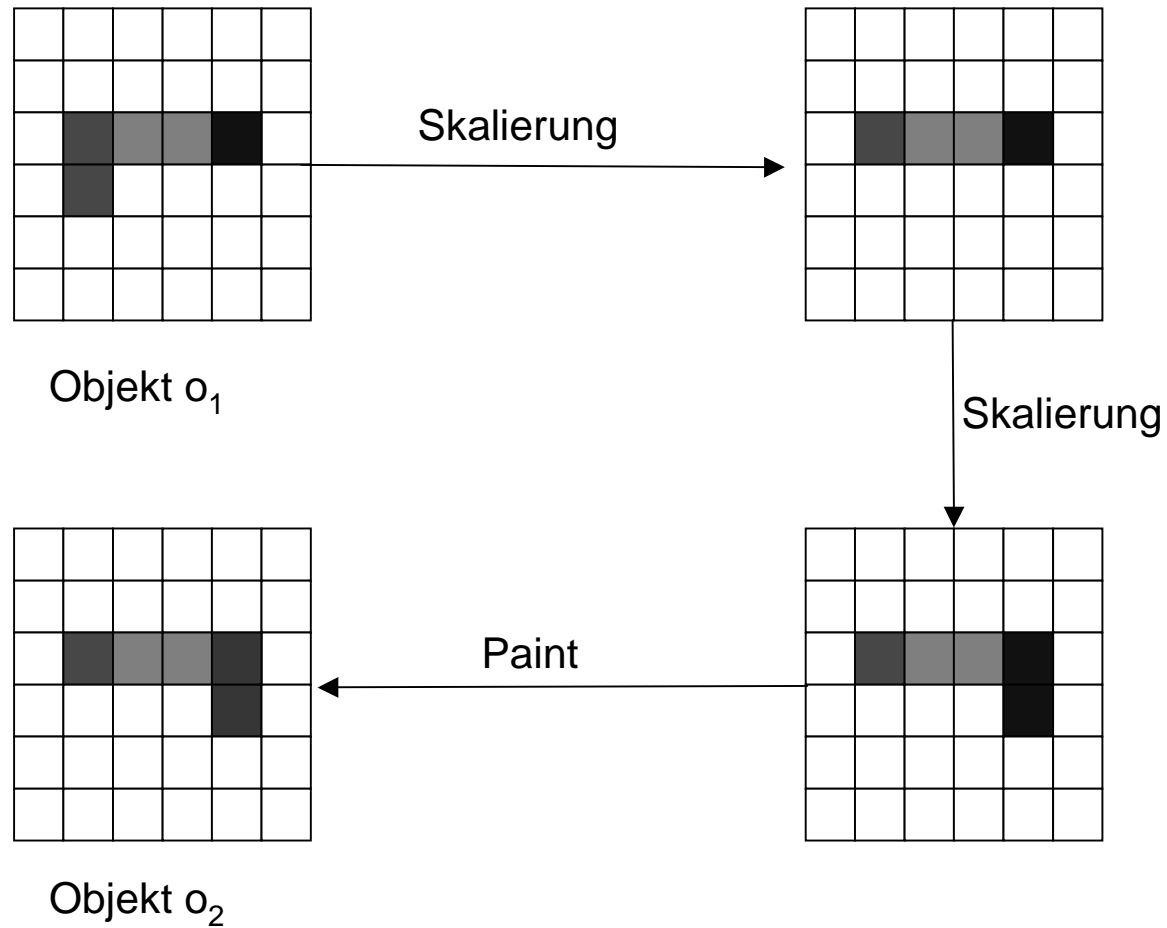
Dieser Ansatz ist allgemeiner als der metrische Ansatz und er geht von der Grundlage aus, dass der Unterschied zwischen 2 Objekten o_1 und o_2 proportional zu den (minimalen) Kosten für eine Transformation von o_1 in o_2 oder andersherum ist.

- Transformationsoperatoren to_1, \dots, to_n
- Jedem Operator werden bestimmte Kosten zugeteilt
- Kosten für die Transformation eines Objektes o_1 in ein Objekt o_2 ist die Summe der Einzelkosten

Transformationsbeispiel



Transformationsbeispiel



Transformationsbeispiel

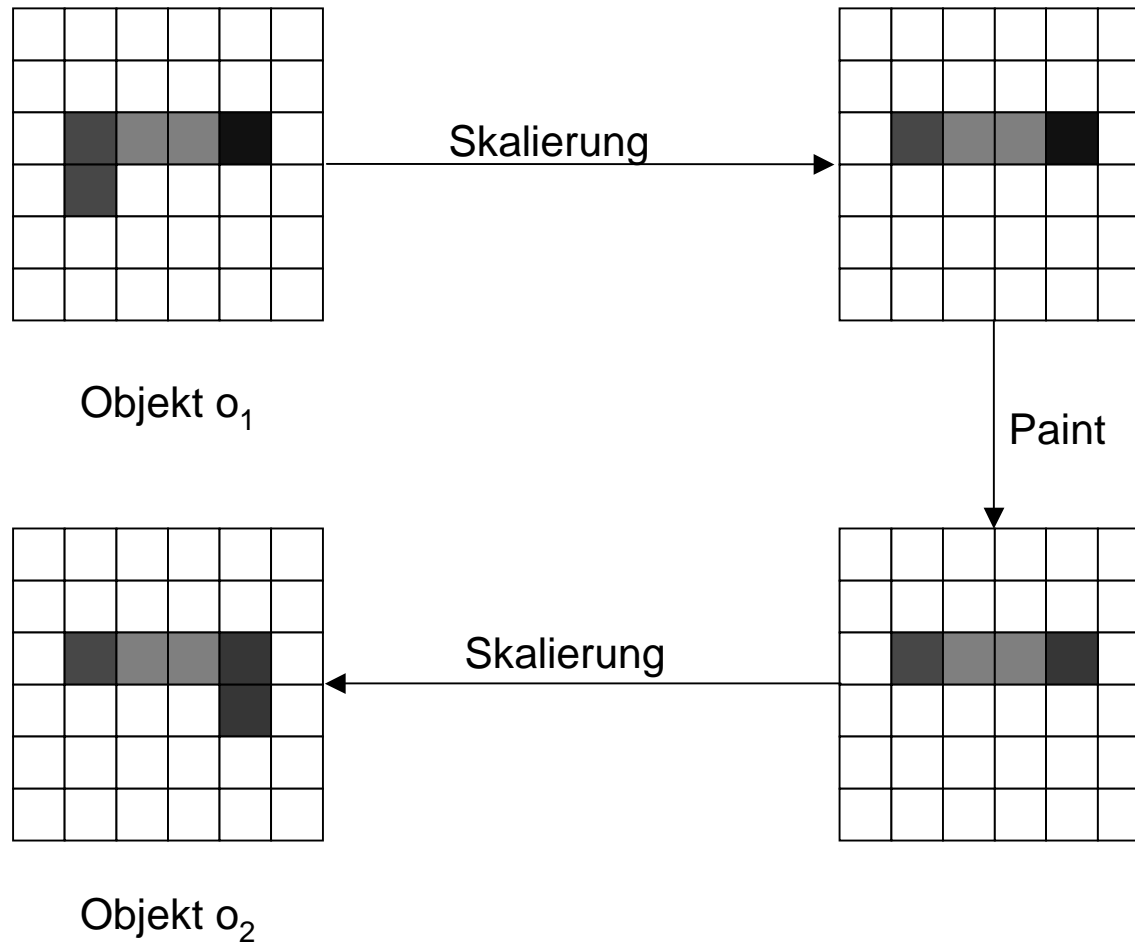


Image Database

- Darstellung einer IDB durch Relationen am Beispiel einer Passphotodatenbank
 - Generierung einer Relation *Image* mit folgendem Schema:
(Image, ObjID, XLB, XUB, YLB, YUB)

Image	Objld	XLB	XUB	YLB	YUB	Image	Objld	XLB	XUB	YLB	YUB
pic1.gif	o ₁	10	60	5	50	pic5.gif	o ₆	0	40	15	50
pic1.gif	o ₂	80	120	20	55	pic6.gif	o ₇	20	75	15	80
pic2.gif	o ₃	20	65	20	75	pic6.gif	o ₈	20	70	130	185
pic3.gif	o ₄	25	75	10	60	pic7.gif	o ₉	15	70	15	75
pic4.gif	o ₅	20	60	30	80						

Image Database



Bild 1



Bild 2



Bild 3



Bild 4



Bild 5



Bild 6



Bild 7

Image Database

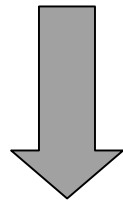
- SQL Anfrage:

Select image

From images I, name N

Where I.ObjId = n.ObjId AND

N.name = "Bjoern Sanders"



Ergebnis

ObjId	Name
o ₁	Bjoern Sanders
o ₄	Bjoern Sanders

Image Database

ObjId	Name	Wahrscheinlichkeit
o ₁	Bjoern Sanders	0.8
o ₁	Jens Wrede	0.2
o ₂	Dirk Niebuhr	0.75
o ₂	Jens Wrede	0.25
o ₃	Dirk Niebuhr	1
o ₄	Bjoern Sanders	1
o ₅	Judith Werner	1
o ₆	Sabine Queckbörner	1
o ₇	Judith Werner	0.7
o ₇	Sabine Queckbörner	0.3
o ₈	Sybille Hardt	0,65
o ₈	Sabine Queckbörner	0,30
o ₈	Bjoern Sanders	0,05
o ₉	Sybille Hardt	1

Anwendung
eines
Bilderkennungs-
programms

Image Database

ObjId	Name	Wahrscheinlichkeit
o_{10}	Judith Werner	0.5
o_{10}	Sabine Queckbörner	0.5
o_{11}	Bjoern Sanders	0.8
o_{11}	Jens Wrede	0.2



Zusätzliches Bild

Wie berechnet sich die Wahrscheinlichkeit,
dass Bild acht Judith Werner und Björn Sanders zeigt ?

➔ Reine Multiplikation der Wahrscheinlichkeiten reicht nicht aus

↳ Folgende Gleichungssysteme müssen gelöst werden

Image Database

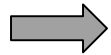
Fall 1: o_{10} ist Judith Werner und o_{11} ist Bjoern Sanders

Fall 2: o_{10} ist Judith Werner und o_{11} ist nicht Bjoern Sanders

Fall 3: o_{10} ist nicht Judith Werner aber o_{11} ist Bjoern Sanders

Fall 4: o_{10} ist nicht Judith Werner und o_{11} ist nicht Bjoern Sanders

Image Database



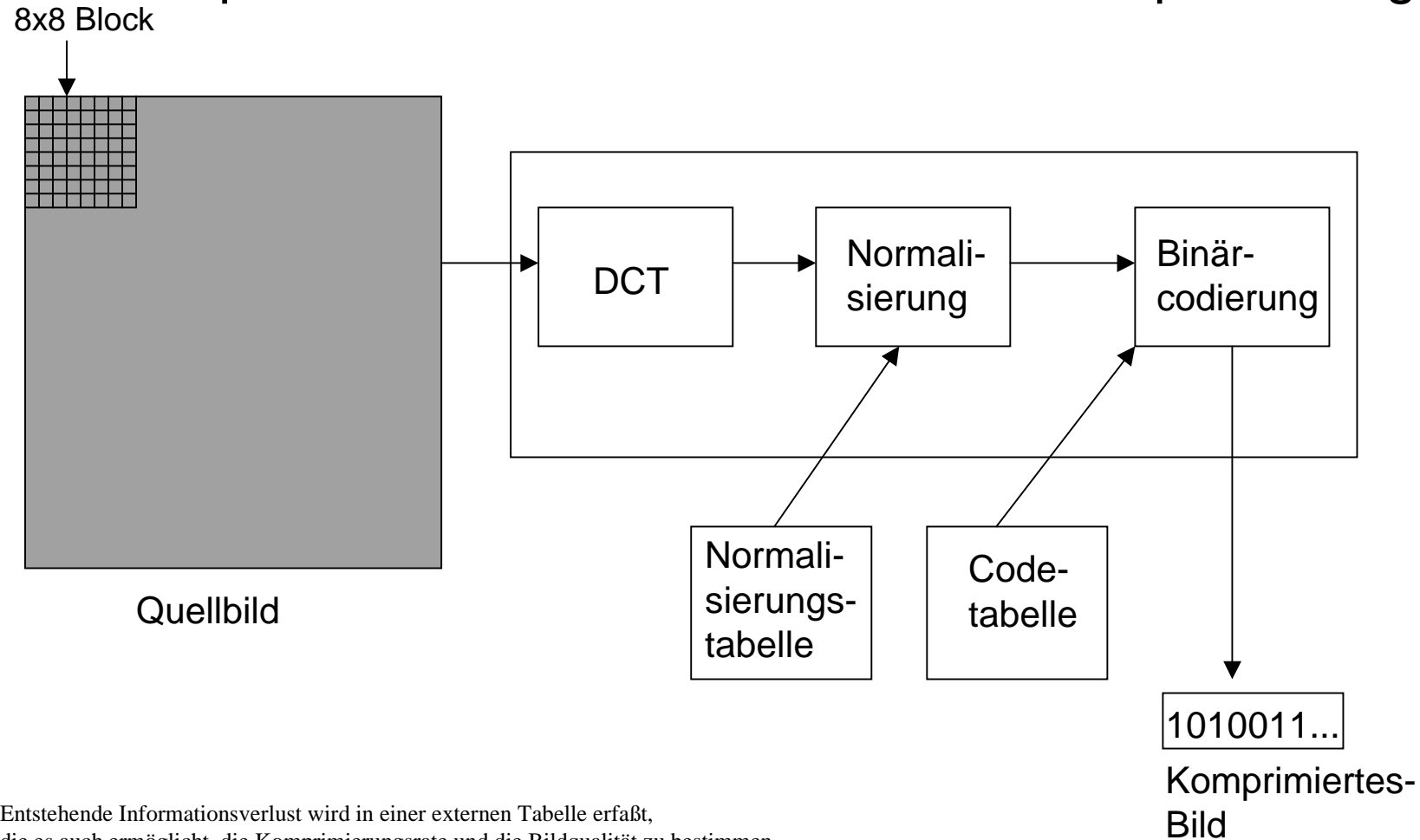
ObjId	Name	Prob(Lower)	Prop(Upper)
o_1	Bjoern Sanders	0.77	0.83
o_1	Jens Wrede	0.17	0.23
o_2	Dirk Niebuhr	0.72	0.78
o_2	Judith	0.22	0.28
o_3	Dirk Niebuhr	0.97	1
o_4	Bjoern Sanders	0.97	1
o_5	Judith Werner	0.97	1
o_6	Sabine Queckbörner	0.97	1
o_7	Judith Werner	0.67	0.73
o_7	Sabine Queckbörner	0.27	0.33
o_8	Sybille Hardt	0.62	0.68
o_8	Sabine Queckbörner	0.27	0.33
o_8	Bjoern Sanders	0.02	0.08
o_9	Sybille Hardt	0.97	1
o_{10}	Judith Werner	0.47	0.53
o_{10}	Sabine Queckbörner	0.47	0.53
o_{11}	Bjoern Sanders	0.77	0.83
o_{11}	Jens Wrede	0.17	0.23

**NOCH IRGENDWELCHE
FRAGEN ?**

- **Diskette:** 1.44 Mbyte, 1,423 Kbyte, 1.457.152 Byte \longrightarrow 11.657.216 Bit
 - Normaler **Text:** 1 **Seite:** 64 Zeilen, **Zeile:** 80 Zeichen und ein Zeichen: 1 Byte groß
 - **Datenvolumen:** 80 Byte/Zeile x 64 Zeilen/Seite x 8 = 40960 Bit/Seite
- Es passen $11.657.216 \text{ Bit} / 40960 \text{ Bit/Seite} = 284,6$ Seiten Text auf eine Diskette
 - Standardbild mit 512x512 **Bildpunkten**, Bildpunkt besitze Farbtiefe von 24 Bit (je acht Bit für R,G,B)
 - **Bildgröße:** 512x512 Bildpunkte/Bild x 24 Bit/Bild = 6.291.456 Bit/Bild.
- $11.657.216 \text{ Bit} / 6.291.456 \text{ Bit/Bild} = 1,9$ Bilder pro Diskette

Bildkomprimierung

Sequentielle DCT-basierende JPEG-Komprimierung



Entstehende Informationsverlust wird in einer externen Tabelle erfaßt, die es auch ermöglicht, die Komprimierungsrate und die Bildqualität zu bestimmen.

Eine weitere externe Tabelle wird für die Codierung benötigt. Diese Tabelle muß die Erzeugung der Codes steuern.

Wenn diese beiden Tabellen bei der Speicherung des komprimierten Bildes fehlen sollten, ist eine Dekomprimierung nicht mehr möglich.

diskrete Cosinustransformation

Quantisierung

Quantisierung: Übergang von der Beschreibung einer Frequenz durch ein großes Alphabet zu der Beschreibung mit einem kleinen Alphabet

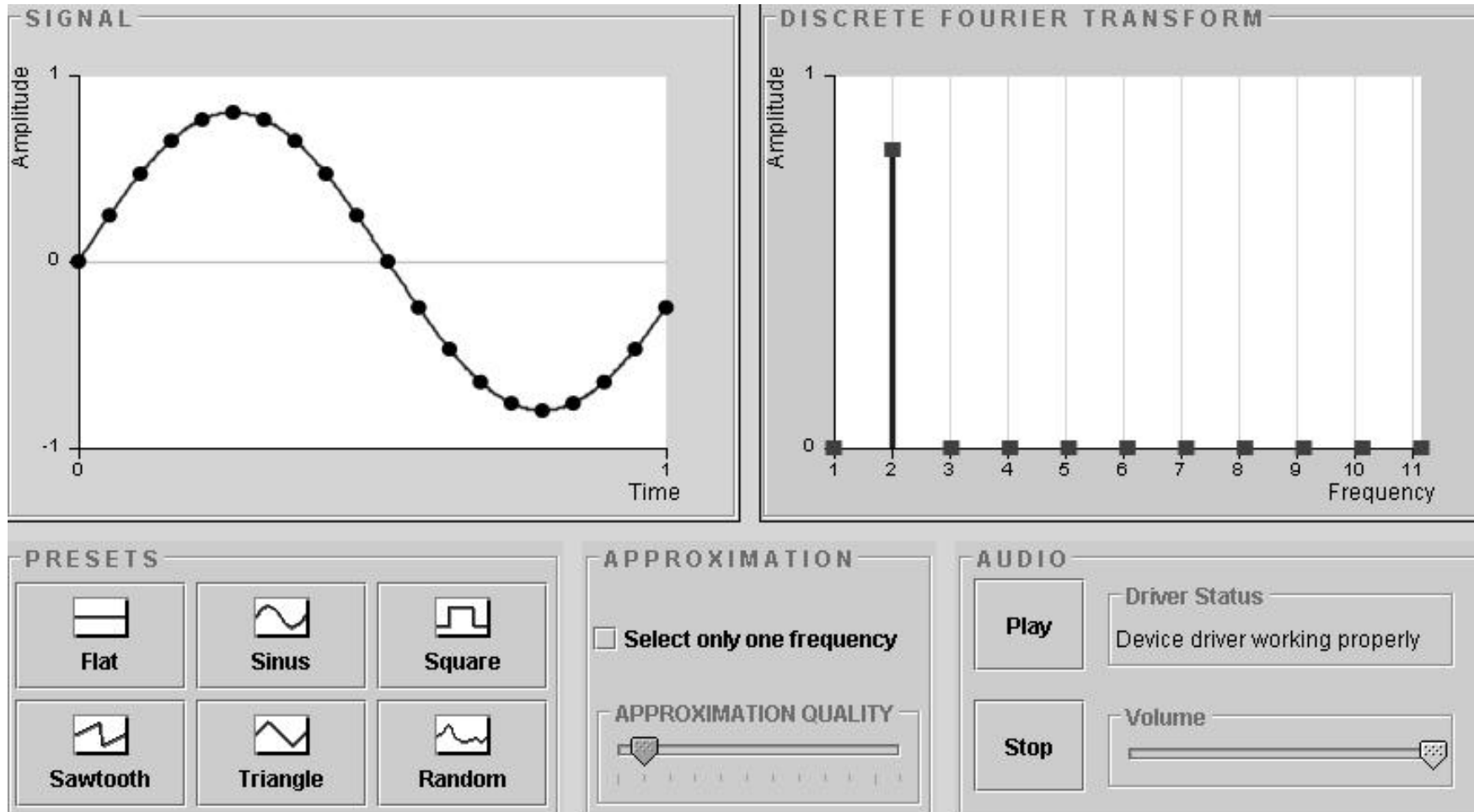
Beispiel: Ein Grauwertbild sei mit 8 bit/Pixel gespeichert.

D.h. jedem Pixel stehen 256 Grauwerte zur Verfügung

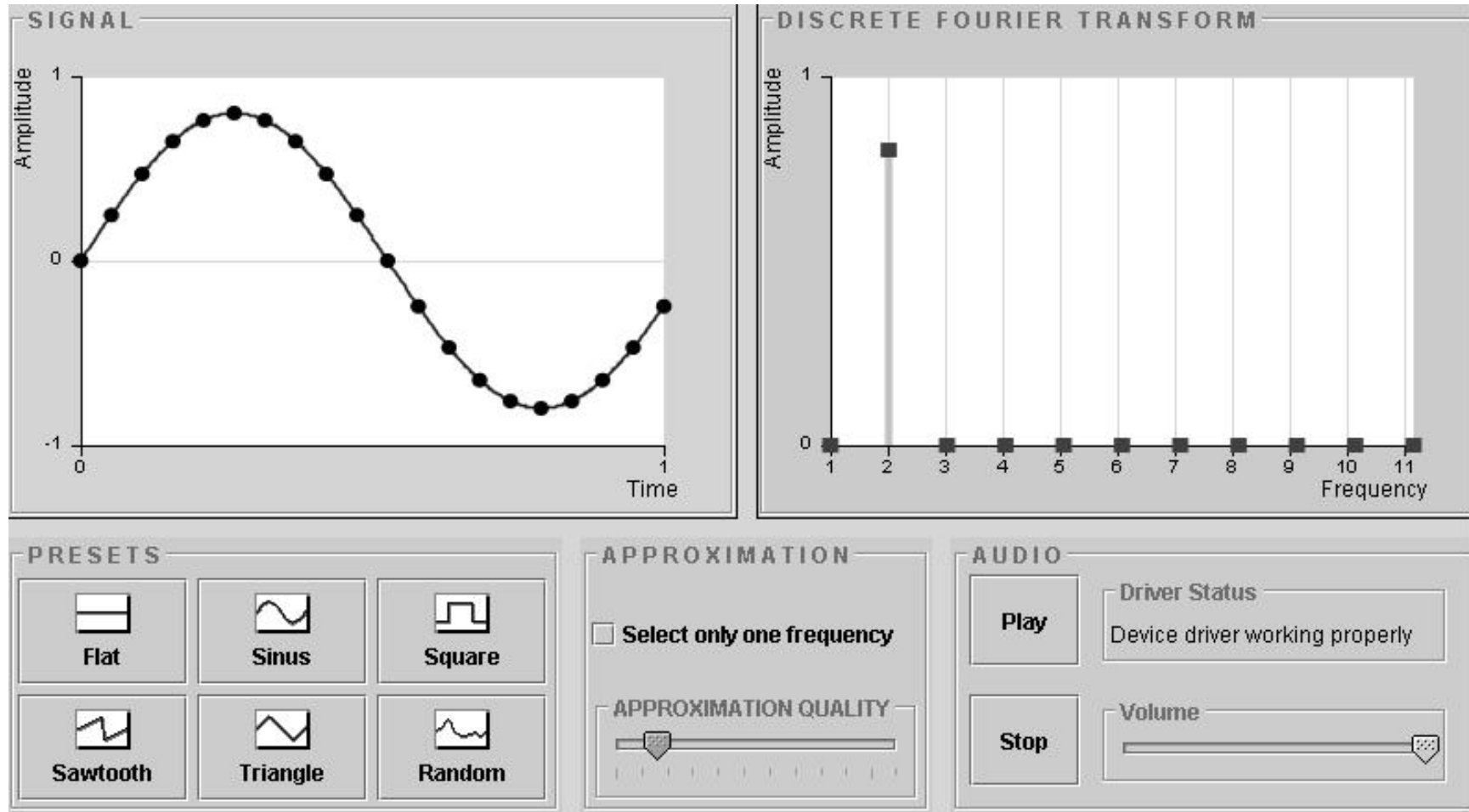
Quantifizierung auf 4 bit/Pixel stehen nur noch 16 Grauwerte zur Verfügung

Ausgangsbild kann nicht mehr exakt hergestellt werden!!!!

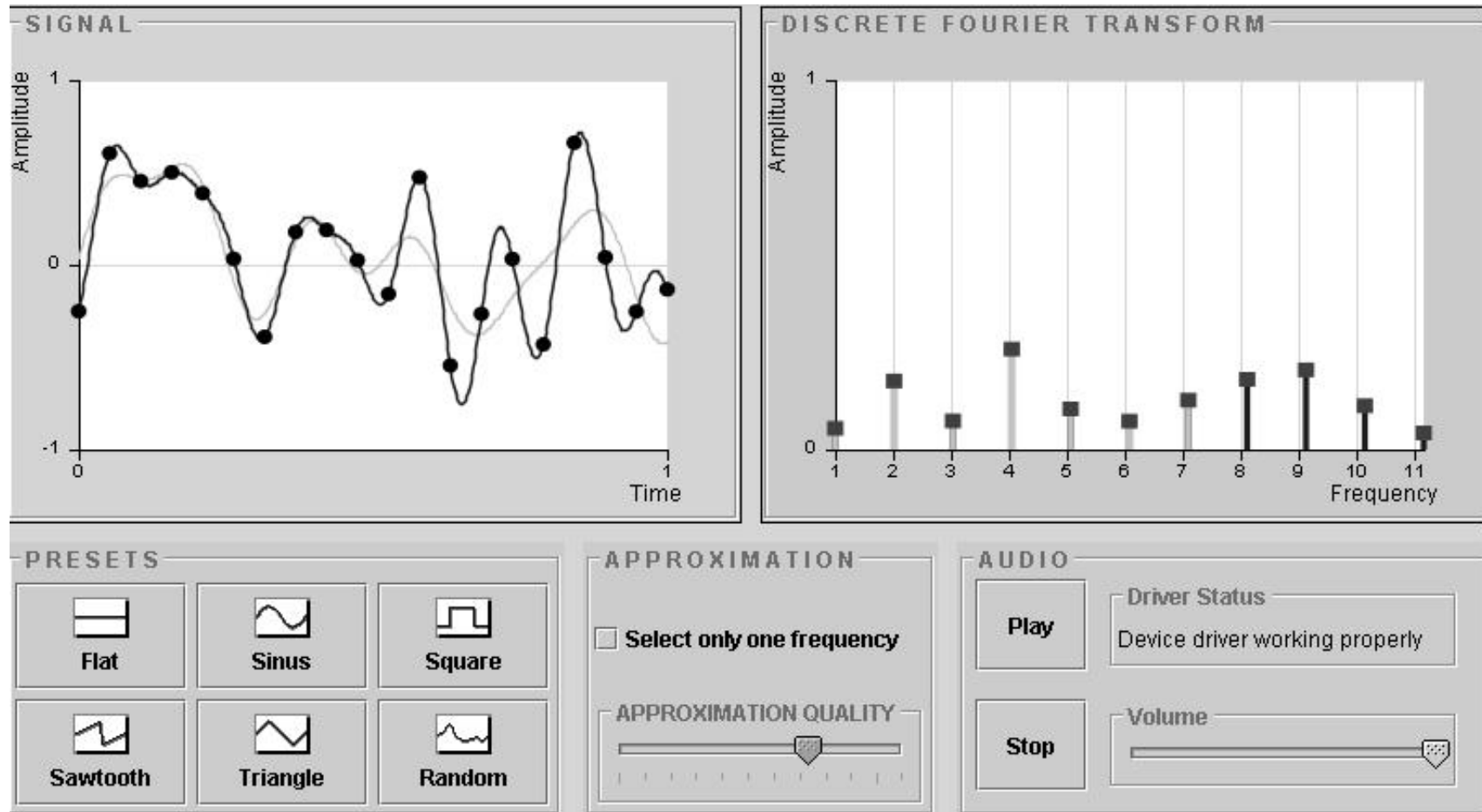
Fourier Transformation



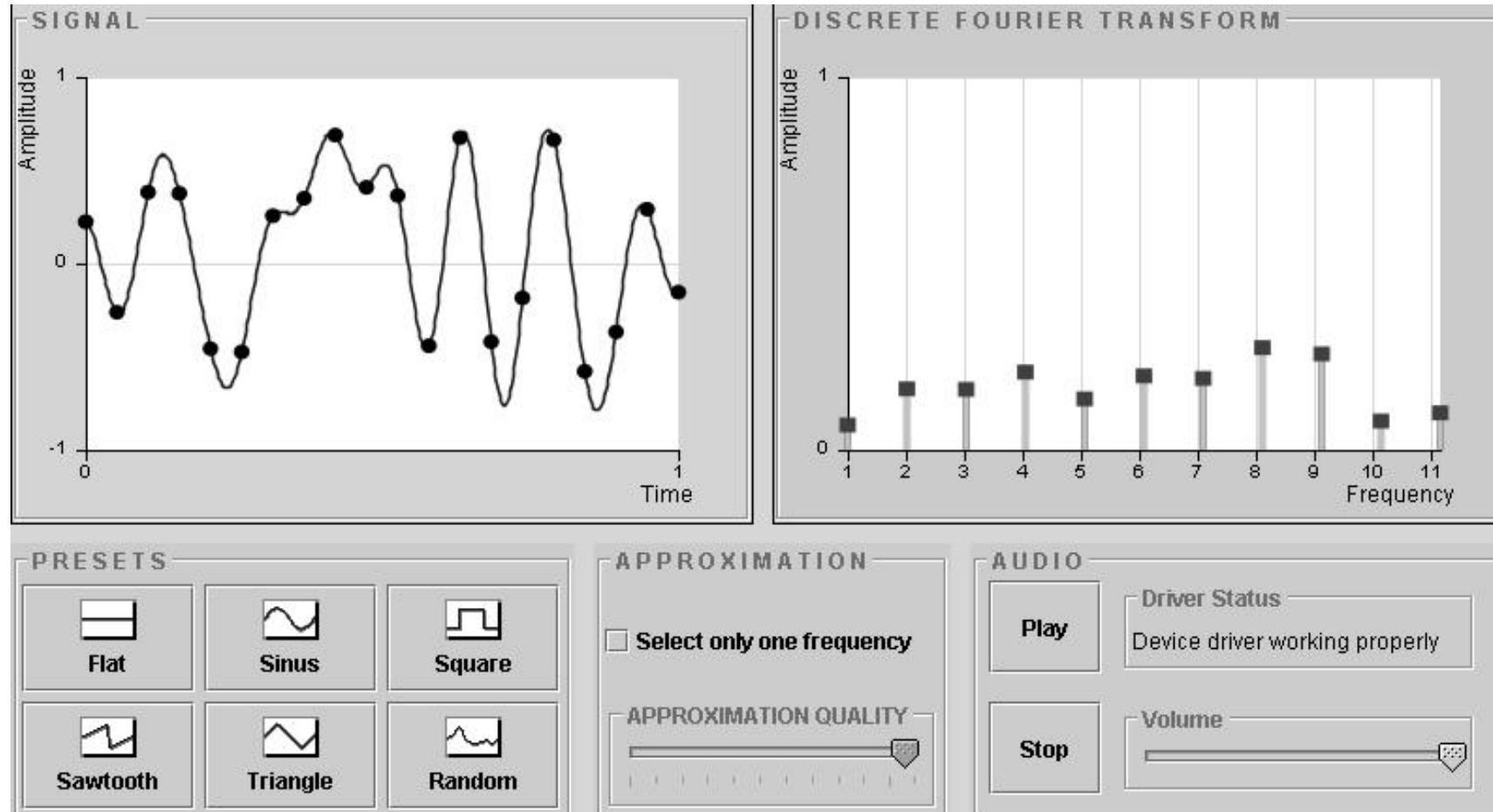
Fourier Transformation



Fourier Transformation



Fourier Transformation



Fourier Transformation