Datenkompression:

Transformationscodierungen

H. Fernau

email: fernau@uni-trier.de

WiSe 2008/09
Universität Trier

Transformation für Codierungen: Das Schema

Seien $x_1, x_2, \ldots, x_k$ Größen, wie z.B. Zahlen, Vektoren, Funktionen etc. Dann ändern wir diese Objekte mit Hilfe einer *Transformation T* und bekommen $\theta_1, \theta_2, \ldots, \theta_t$, die möglicherweise ganz anders als x_i aussehen, die man aber einfach weiterverarbeiten kann.

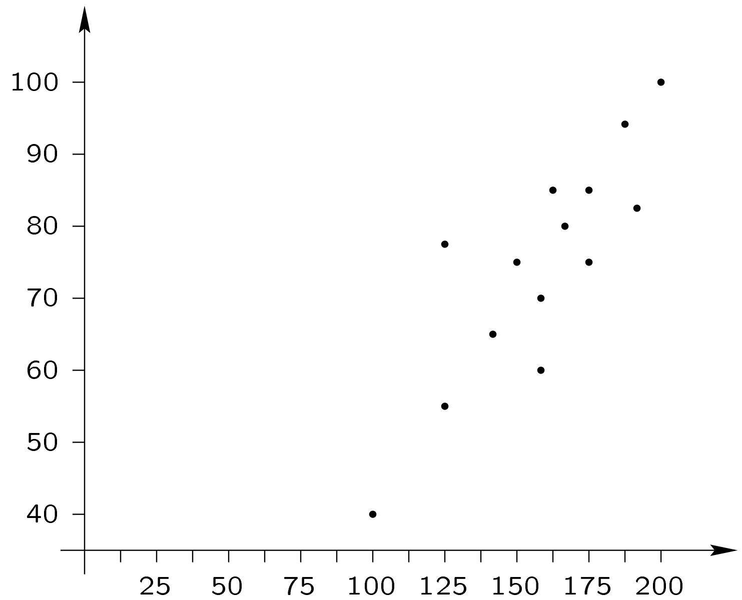
$$\begin{tikzcd}
x_1, x_2, \ldots, x_k \arrow[r, "\textit{Transformation}"] & \theta_1, \theta_2, \ldots, \theta_t \arrow[d, "\textit{Bearbeitung}"] \\
\widehat{x}_1, \widehat{x}_2, \ldots, \widehat{x}_k & \widehat{\theta}_1, \widehat{\theta}_2, \ldots, \widehat{\theta}_t \arrow[l, "\textit{inverse Transformation}"']
\end{tikzcd}$$

Ziel: Transformationen zu finden, sodass man $\theta_1, \theta_2, \ldots, \theta_t$ effizienter (verlustbehaftet) komprimieren kann als die Folge $x_1, x_2, \ldots, x_k$ direkt.
Dann ist $\widehat{\theta}_1, \widehat{\theta}_2, \ldots, \widehat{\theta}_t$ eine Rekonstruktion dee Komprimierung der Folge $\theta_1, \theta_2, \ldots, \theta_t$, und $\widehat{x}_1, \widehat{x}_2, \ldots, \widehat{x}_k$ ist eine Rekonstruktion der Eingabe $x_1, x_2, \ldots, x_k$.
Bekannte Beispiele: Wavelets, BWT, . . .

Ein Beispiel-Eingangsdatensatz:

Größe	Gewicht
163	85
188	94
150	75
175	85
140	65
200	102
170	80
125	55
100	40
125	77
173	74
155	70
190	82
160	60

Graphische Darstellung:



Die **konkrete Transformation**: Betrachtet man Länge und Gewicht als (x, y)-Koordinaten, so sieht man im Bild, dass sich die Daten längs einer Geraden der Art $y = 0,5x$ häufen. Drehen wir den Datensatz also durch $\theta = \mathbf{A}\mathbf{x}$, wobei

$$\mathbf{x} = \left[\begin{array}{c} x_0 \\ x_1 \end{array} \right]$$

der zweidimensionale Eingabevektor ist und

$$\mathbf{A} = \left[\begin{array}{cc} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{array} \right]$$

die Drehmatrix sowie

$$\theta = \left[\begin{array}{c} \theta_0 \\ \theta_1 \end{array} \right]$$

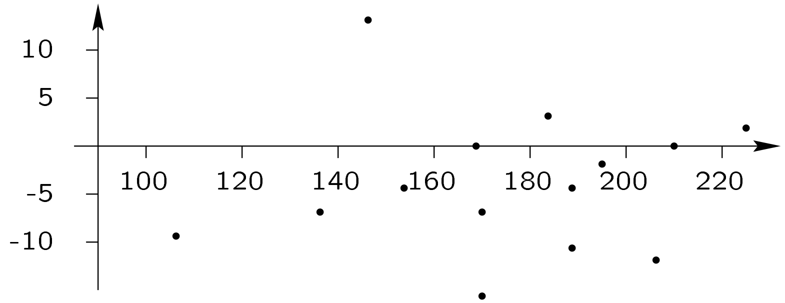
der Vektor der Transformierten; speziell ist hier $\phi = \arctan(0,5) \approx 26,565^o$, d. h.

$$\mathbf{A} = \left[\begin{array}{cc} 0,89442719 & 0,4472136 \\ -0,4472136 & 0,89442719 \end{array} \right]$$

Die transformierten Werte

x-Wert	y-Wert
184	3
210	0
168	0
195	-2
154	-4
225	2
188	-4
136	-7
107	-9
146	13
188	-11
170	-7
207	-12
170	-18

Die transformierten Werte als Graphik



Die rücktransformierten Werte

Größe	Gewicht
165	82
188	94
150	75
174	87
138	69
201	101
168	84
122	61
96	48
131	65
168	84
152	76
185	93
152	76

Beobachtungen

Wir beobachten bei den transformierten Werten, dass sich die „Energie" in der ersten Komponente „ballt", was intuitiv eine (verlustbehaftetete) Komprimierung um den Faktor zwei gestattet. Täten wir dies tatsächlich und wendeten auf die „komprimierte Folge" die inverse Transformation, gegeben durch

$$\mathbf{A}^{-1} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

an, so erhielten wir in etwa die Werte aus der vorigen Tabelle.

In diesem Fall gilt:

$$\sum_{i=0}^{N-1}(x_i - \widehat{x}_i)^2 = \sum_{i=0}^{N-1}(\theta_i - \widehat{\theta}_i)^2$$

wobei

$$\widehat{\theta}_i = \begin{cases} \theta_i & i = 0, 2, 4, \ldots \\ 0 & \text{sonst} \end{cases}$$

und $\widehat{x}_i$ der x_i entsprechende rekonstruierte Wert ist. Diese Eigenschaft gilt allgemein für Transformationsmatrizen mit $A^{-1} = A^T$. Letztere Eigenschaft nennt man auch *Orthonormalität*.

Einfache Transformationen

Zuerst betrachten wir Transformationen für eindimensionale Folgen, wie sie zum Beispiel (digitalisierte) Sprache und allgemeiner Audio-Folgen liefern. Es sei $p_0, p_1, p_2, \ldots$ die Eingabefolge. Wir werden folgende *lineare Transformationen* betrachten: Für jeden Block

$$x_0 = p_\ell,\ x_1 = p_{\ell+1}, \ldots, x_{N-1} = p_{\ell+N-1}$$

der Länge N, mit $\ell = 0, N, 2N, \ldots$, transformieren wir $x_0, x_1, \ldots, x_{N-1}$ folgendermaßen:

$$\theta_n = \sum_{i=0}^{N-1} x_i a_{n,i},$$

wobei die Transformation vollständig durch die Konstanten $a_{n,i}$ definiert wird. Die Länge N hängt von praktischen Anwendungen ab. Es ist klar, dass mit größerem N auch die Transformation komplexer wird. Seien nun x und θ die Vektoren. Dann können wir eine lineare Transformation in Matrixform darstellen: $\theta = Ax$. Die inverse Transformation definiert eine Matrix B, so dass $x = B\theta$ gilt. Die Matrix B ist die Inverse zu A.

Der zweidimensionale Fall:
Transformationscodierung ist eine der populärsten Methoden für die Komprimierung von Bildern.
$\leadsto$ Wir werden bis Ende dieses Kapitels meistens den zweidimensionalen Fall der linearen Transformation betrachten. Wir definieren solche Transformationen für einen gegebenen $N \times N$ Block

$$X = \begin{bmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,N-1} \\ x_{1,0} & x_{1,1} & \cdots & x_{1,N-1} \\ \vdots & & & \vdots \\ x_{N-1,0} & x_{N-1,1} & \cdots & x_{N-1,N-1} \end{bmatrix}$$

folgendermaßen:

$$\Theta = AXA^T$$

und die inverse Transformation: $X = B\Theta B^T$. Alle Transformationen, die wir hier betrachten werden, sind orthonormal. Daher ist es einfach, die inverse Transformation zu bekommen:

$$X = A^T \Theta A$$

Der zweidimensionale Fall (Forts.):

Die Zeilen der Transformationsmatrix nennen wir *Basisvektoren* und die Elemente nach der Transformation nennen wir oft *Transformationskoeffizienten*.

Man beachte, dass wir auf diese Art und Weise nur sog. *separable Transformationen* für den zweidimensionalen Fall erhalten; denn die allgemeine Transformationsformel lautet

$$\Theta_{k,\ell} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{i,j} a_{i,j,k,\ell},$$

(dies wäre ein Tensorprodukt) und wir betrachten als Spezialisierung:

$$\Theta_{k,\ell} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{i,j} a_{k,i} a_{\ell,j}.$$

Die **Wirksamkeit einer Transformation** hängt davon ab,

- wie groß die *Dekorrelation*, also die Reduktion der Korrelation, der Eingabepixel (oder Signale) ist und
- wieviel *Energiebündelung* (engl.: energy compaction) die Transformation ergibt.

Die Korrelation in einer Eingabefolge ist die Quelle der Redundanz, die in einem effizienten Komprimierungsverfahren entfernt werden soll.

Eine wichtige Eigenschaft der orthonormalen Transformationen ist, dass solche Transformationen die Energie erhalten:

$$\sum x_{i,j}^2 = \sum \theta_{i,j}^2.$$

Die Transformationen sollen nur die Energie „nach links oben" in der Matrix verschieben, also dort bündeln.

Beispiel: Betrachten wir die folgende Transformationsmatrix:

$$A = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

A ist orthonormal, weil

$$\frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^T = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^T \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Die erste Zeile der Matrix $(1/\sqrt{2}, 1/\sqrt{2})$ enspricht einem Tiefpass und die zweite Zeile $(1/\sqrt{2}, -1/\sqrt{2})$ enspricht einem Hochpass.

Betrachten wir eine Folge, in der beide Elemente gleich sind. Nach der Transformation soll dann das zweite Element Null sein, d. h., die Energie ist vollständig in der ersten Komponente gebündelt. Es sei (α, α) eine solche Eingabefolge. Es gilt:

$$\theta = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \alpha \end{bmatrix} = \begin{bmatrix} \sqrt{2}\alpha \\ 0 \end{bmatrix}$$

Der zweidimensionale Fall

Beispiel: Wir nehmen die selbe Transformationsmatrix wie im vorigen Beispiel.

$$A = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Eingabematrix: $\begin{bmatrix} \alpha & \alpha \\ \alpha & \alpha \end{bmatrix} \rightsquigarrow$ Transformierte:

$$\Theta \;=\; \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \alpha & \alpha \\ \alpha & \alpha \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^T \;=\; \begin{bmatrix} 2\alpha & 0 \\ 0 & 0 \end{bmatrix}$$

Beobachtung: Die ganze Energie ist im linken oberen Eck der Matrix konzentriert.

Charakteristika der transformierten Matrizen

Aus historischen Gründen heißt der $\theta_{0,0}$-Koeffizient (im linken oberen Eck der Matrix der Transformierten Werte) *DC-Koeffizient* oder *Niederfrequenz-Koeffizient* und die anderen *AC-Koeffizienten* oder *Hochfrequenz-Koeffizienten*. DC steht für *direct current*, zu deutsch Gleichstrom, und AC steht für *alternating current* (Wechselstrom). Einen Grund für diese Namensgebung sieht man, wenn man eine inverse Transformation für A betrachtet:

$$X = A\Theta A^T = \frac{1}{2}\begin{bmatrix} \theta_{0,0}+\theta_{0,1}+\theta_{1,0}+\theta_{1,1} & \theta_{0,0}-\theta_{0,1}+\theta_{1,0}-\theta_{1,1} \\ \theta_{0,0}+\theta_{0,1}-\theta_{1,0}-\theta_{1,1} & \theta_{0,0}-\theta_{0,1}-\theta_{1,0}+\theta_{1,1} \end{bmatrix}$$

Das heißt:

$$X = \theta_{0,0}\alpha_{0,0}+\theta_{0,1}\alpha_{0,1}+\theta_{1,0}\alpha_{1,0}+\theta_{1,1}\alpha_{1,1},$$

wobei

$$\alpha_{0,0} = \tfrac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \qquad \alpha_{0,1} = \tfrac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

$$\alpha_{1,0} = \tfrac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \qquad \alpha_{1,1} = \tfrac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Beobachten Sie, dass alle Elemente der $\alpha_{0,0}$ Matrix gleich sind. Daher kommt die DC-Benennung.

Karhunen-Loéve-Transformation KLT

KLT ist nachweislich optimal hinsichtlich zweier Kriterien:

- Sie dekorreliert die Daten am besten und
- sie bündelt damit am besten die Information in den niederfrequenten Bereich.

Die entsprechenden mathematischen Messgrößen η_C und η_E werden wir weiter unten einführen.

Die wichtigsten Anfangsschritte der Transformation sind:

1. Berechne Kovarianzmatrix $COV(X)$ der Daten und
2. bestimme die Eigenvektoren

jeweils für jedes Bild oder für jede Bildgruppe, da die Basisvektoren nur für die jeweilige Datenmenge chrakteristisch sind.
Zusätzlich müssen noch die Basisvektoren zusammen mit den Kompressionseingaben als Begleitinformationen verschickt werden.
Deshalb ist diese Methode in der Praxis *nicht* nützlich, stellt aber doch eine wichtige Messlatte für die Güte von Kompressionsverfahren dar.

Karhunen-Loéve-Transformation KLT auf Bildern

Wir erinnern zunächst an die Begriffe *Mittelwert* $\bar{x}$ und *Varianz* σ^2 für das ganze Bild ($N \times M$ Pixel):

$$\bar{x} = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} x_{i,j}$$

$$\sigma^2 = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (x_{i,j} - \bar{x})^2.$$

Jetzt betrachten wir –als Vorbereitung für den zweidimensionalen Fall– die Korrelation im Eindimensionalen.

Für eine (eindimensionale) Folge x ist die Varianz:

$$\sigma^2 = \sigma_{xx}^2 = E[(x - \bar{x})^2].$$

Für zwei Folgen x und y definieren wir als ihre *Kovarianz*

$$\sigma_{xy}^2 = E[(x - \bar{x})(y - \bar{y})].$$

Hier und im Folgenden werden wir als „Erwartungswert" stets den Mittelwert ansetzen, was ja auch ein üblicher Schätzer ist. Ähnlich „lässig" verfahren wir mit anderen Begriffen aus Wahrscheinlichkeitstheorie und Statistik.

Ist $\sigma_{xy}^2 = 0$, so sind die beiden Folgen *unkorreliert*.

Kovarianzmatrix I

Jetzt betrachten wir als Beispiel drei Folgen x, y und z. Dann erhalten wir die folgende *Kovarianzmatrix*:

$$COV(X) = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 & \sigma_{xz}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 & \sigma_{yz}^2 \\ \sigma_{zx}^2 & \sigma_{zy}^2 & \sigma_{zz}^2 \end{bmatrix}$$

Für k Folgen ist $COV(X)$ analog eine $k \times k$-Matrix.

Jetzt definieren wir für die Folge

$$x_1, x_2, \ldots, x_M$$

als *k-te Kovarianz*

$$\sigma_{1,k+1}^2 = \sigma_{k+1,1}^2 = \frac{1}{M-k} \sum_{i=1}^{M-k} (x_i - \bar{x})(x_{i+k} - \bar{x}),$$

d. i. die Kovarianz für die Folgen $x_1, x_2, \ldots, x_{M-k}$ und $x_{k+1}, x_{k+2}, \ldots, x_M$, wobei wir als Mittelwert für beide Folgen (vereinfachend) $\bar{x}$ nehmen.
Vereinfachend setzen wir weiter:

$$\sigma_{1,k+1}^2 = \sigma_{k+1,1}^2 = \frac{1}{M} \sum_{i=1}^{M-k} (x_i - \bar{x})(x_{i+k} - \bar{x}).$$

Für $k \leq M$ ist das immer eine gute Abschätzung.

Kovarianzmatrix II

Dann bekommen wir:

$$\sigma_{11}^2 = \sigma_{22}^2 = \ldots \sigma_{kk}^2 = \sigma^2$$

$$\sigma_{12}^2 = \sigma_{23}^2 = \ldots \sigma_{k-1,k}^2 = \sigma_1^2$$

usw., wobei wir $\sigma_{\ell,k}^2$ ähnlich definieren wie $\sigma_{1,k+1}^2$ nur für die Folgen, die mit x_ℓ bzw. $x_{\ell+k}$ starten. Dann ist die *Kovarianzmatrix einer Folge*:

$$COV(X) = \begin{bmatrix} \sigma^2 & \sigma_1^2 & \sigma_2^2 & \ldots & \sigma_{k-1}^2 \\ \sigma_1^2 & \sigma^2 & \sigma_1^2 & \ldots & \sigma_{k-2}^2 \\ \vdots & \vdots & \vdots & & \vdots \\ \sigma_{k-1}^2 & \sigma_{k-2}^2 & \sigma_{k-3}^2 & \ldots & \sigma^2 \end{bmatrix}$$

Wir normalisieren die Matrix:

$$COV(X) = \sigma^2 \begin{bmatrix} 1 & \rho_1 & \rho_2 & \ldots & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \ldots & \rho_{k-2} \\ \vdots & \vdots & \vdots & & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \ldots & 1 \end{bmatrix},$$

wobei $\rho_k = \sigma_k^2/\sigma^2$ der *k-te Korrelationskoeffizient* ist.

Kovarianzmatrizen in der Bildverarbeitung

Zur Bildverarbeitung passt der folgende Fall erfahrungsgemäß sehr gut:*

$$\rho_k = \rho^k,$$

mit $\rho = \rho_1$. Wir nennen ρ *Zwischen-Element-Korrelation*. Dann definieren wir die *Korrelationsmatrix*

$$COR(X) = \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{k-1} \\ \rho & 1 & \rho & \dots & \rho^{k-2} \\ \vdots & \vdots & \vdots & & \vdots \\ \rho^{k-1} & \rho^{k-2} & \rho^{k-3} & \dots & 1 \end{bmatrix}$$

Für den zweidimensionalen Fall verallgemeinern sich diese Begriffe mit Hilfe von vertikalen und horizontalen Folgen. Die Verallgemeinerung ist nicht trivial und wir werden sie hier nicht vertiefen.

Daher betrachten wir jetzt nur (noch) den eindimensionalen Fall.

*Genau genommen wird dabei mit der Annahme gearbeitet, die Eingabefolge sei ein stationärer Markov-Prozess k-ter Ordnung. In der Praxis bewährt sich diese an und für sich falsche Annahme.

Dekorrelation und Energiebündelung

Es seien $COV(X) = (X_{i,j})$ und $COV(Y) = (Y_{i,j})$ die Kovarianz-Matrizen für eine eindimensionale Folge und für ihre Transformation. Dann setzen wir:

$$\Sigma X = \sum_{\substack{1 \leq i,j \leq N \\ i \neq j}} |X_{i,j}|$$

und

$$\Sigma Y = \sum_{\substack{1 \leq i,j \leq N \\ i \neq j}} |Y_{i,j}|$$

und definieren damit die *Dekorrelationswirkung* η_C folgendermaßen:

$$\eta_C = 1 - \frac{\Sigma Y}{\Sigma X}.$$

Eine andere Größe, die die Qualität einer Transformation beschreibt, ist der *Energie-Bündelungs-Koeffizient* in den ersten M von N diagonalen Komponenten:

$$\eta_E = \frac{\sum_{j=1,k=j}^{M} Y_{j,k}}{\sum_{j=1,k=j}^{N} Y_{j,k}}.$$

Erinnerung: *Autokorrelationsfunktion* (einer Folge x_i)

$$R_{xx}(k) = E[x_n x_{n+k}]$$

$$\mathbf{R} = \begin{bmatrix} R_{xx}(0) & R_{xx}(1) & R_{xx}(2) & \dots & R_{xx}(N-1) \\ R_{xx}(1) & R_{xx}(0) & R_{xx}(1) & \dots & R_{xx}(N-2) \\ R_{xx}(2) & R_{xx}(1) & R_{xx}(0) & \dots & R_{xx}(N-3) \\ \vdots & \vdots & \vdots & & \vdots \\ R_{xx}(N-1) & R_{xx}(N-2) & R_{xx}(N-3) & \dots & R_{xx}(0) \end{bmatrix}$$

ist die *Autokorrelationsmatrix*.

Beschreibung der KLT

Die Zeilen der Karhunen-Loéve-Transformation sind die Eigenvektoren der *Autokorrelationsmatrix*. Für jedes $i = 0, 1, \ldots, N-1$ lösen wir folgende Gleichung

$$\tan N\omega_i = \frac{-(1-\rho^2)\sin\omega_i}{(1+\rho^2)\cos\omega_i - 2\rho},$$

wobei ρ die Zwischen-Element-Korrelation ist, und dann benutzen wir die Lösungen, um die Eigenvektoren zu berechnen. Dafür setzen wir:

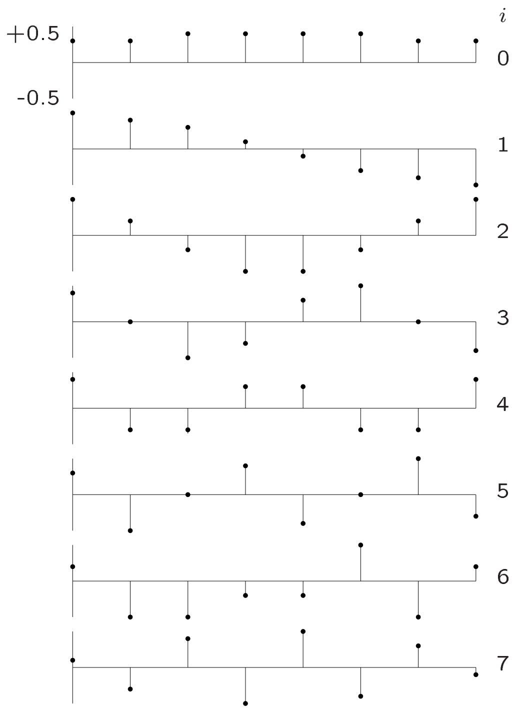
$$\lambda_i = \frac{1-\rho^2}{1+\rho^2-2\rho\cos\omega_i}.$$

Basisvektoren für $0 \leq i, j \leq N-1$ sind dann:

$$a_{i,j} = \left(\frac{2}{N+\lambda_i}\right)^{1/2} \sin\left[\omega_i\left(j - \frac{N-1}{2}\right) + \frac{(i+1)\pi}{2}\right]$$

Intuitiv stellt die Autokorrelationsmatrix gerade die Abhängigkeiten zwischen aufeinander folgenden Gliedern einer Folge dar. Da die Eigenvektoren (als neue Basisvektoren) immer die „Richtungen" angeben, in die Folge „strebt", leistet KLT die gewünschte (optimale) Energiebündelung.

Graphische Darstellung der Werte der acht Basisvektoren der KLT für $N = 8$ und $\rho = 0,91$.



Bit-Verteilung

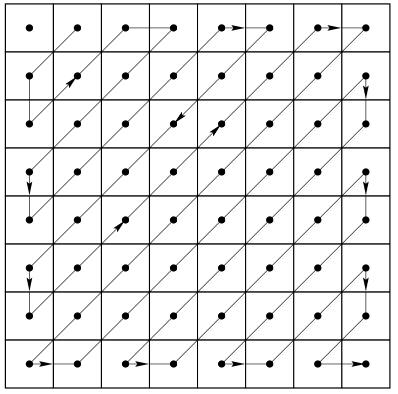
Typische Bitverteilung für eine 8×8-Transformation

8	7	5	3	1	1	0	0
7	5	3	2	1	0	0	0
4	3	2	1	1	0	0	0
3	3	2	1	1	0	0	0
2	1	1	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Wie aus den Beispielen ersichtlich, ist der Informationsgehalt der Einträge der transformierten Matrix unterschiedlich; ähnlich wie bei der zuvor betrachteten Teilbandkompression massiert sich die Energie im linken oberen Eck, und das ist ja auch eines der Kriterien für eine gute Transformation.
Das bedeutet wiederum, dass es sinnvoll ist, bei der quantisierten Übertragung dieser transformierten Koeffizienten unterschiedlich viele Bits zu verwenden. Diese Beobachtung führt zur Anwendung eines schon vorher besprochenen *Bitverteilungsalgorithmus* und wird in diesem Kontext *Zonenabtastung* (engl.: zonal sampling) genannt.

Das **Bitverteilungsverfahren von Chen und Pratt**

Zickzack-Abtastung eines 8×8-Musters



Die zu übertragenden Werte des transformierten Bildes werden zunächt quantisiert und dann in der angegebenen Weise zickzackartig abgetastet.
Man kann annehmen, dass die sich so ergebene abgetastete Zahlenfolge ab einem gewissen Index konstant Null ist (dieser Annahme liegt ja offenkundig die auf der vorigen Folie angegebene fixe Bitverteilung zugrunde); daher erscheint es sinnvoll, eine solche konstante Nullfolge durch ein Sonderzeichen, hier EOB (*Blockende*) genannt, dem Empfänger anzuzeigen.
Vorteil: noch vorhandene „hochfrequente" Informationen nicht automatisch vernachlässigt
Hinweis: auch in JPEG verwendetes Schema
Ähnliche Idee: Shapiros EZW