Département d'informatique et de génie logiciel
Compression de données
IFT-4003/IFT-7023

Notes de cours
Codage par transformation

Édition Hiver 2012

Mohamed Haj Taieb

Local: PLT 2113

Courriel: mohamed.haj-taieb.1@ulaval.ca

Faculté des sciences et de génie
Département de génie électrique et de génie informatique

UNIVERSITÉ LAVAL

Plan de la présentation

Codage par transformation:

- Exemple de motivation
- Processus de transformation
- Transformation KLT
- Transformation DCT
- Quantification des coefficients de la transformation
- Codage des coefficients

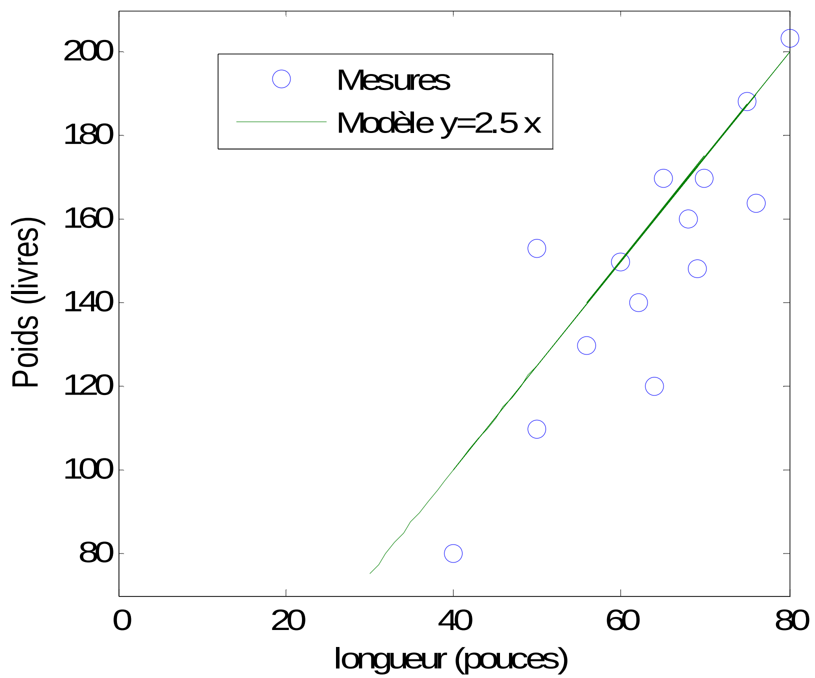
Mohamed Haj Taieb, Département de génie électrique et de génie informatique

Exemple de motivation (1)

❑ Exemple

- Soit un ensemble de mesures de la longueur *X* (en pouces) et du poids *Y* (en livres).
- *X* et *Y* sont fortement corrélés: $y=2.5\ x$.

Height	Weight
65	170
75	188
60	150
70	170
56	130
80	203
68	160
50	110
40	80
50	153
69	148
62	140
76	164
64	120



Exemple de motivation (2)

Rotation

$$\text{Transformation}: \theta = Ax$$

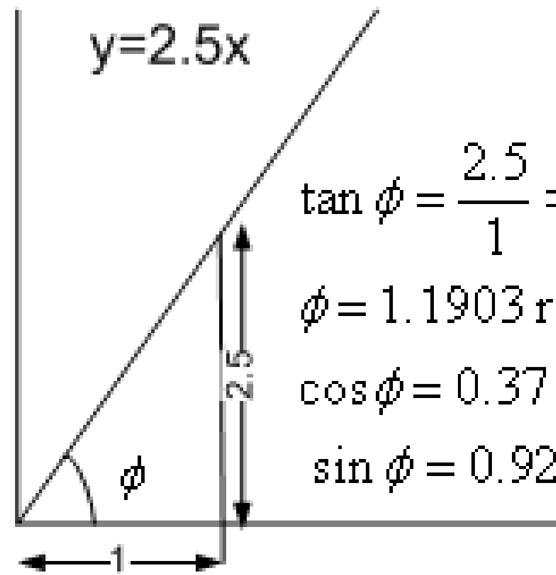
$$\text{Vecteur source: } x = \begin{bmatrix} x_0 \text{ longueur} \\ x_1 \text{ poids} \end{bmatrix}$$

$$\text{Matrice de rotation: } A = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$$

ϕ angle entre l'axe des x et la ligne $y = 2.5x$

$$\text{Résultat de la rotation } \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.37139 & 0.92848 \\ -0.92848 & 0.37139 \end{bmatrix}$$



$$\tan\phi = \frac{2.5}{1} \Rightarrow \phi = \tan^{-1} 2.5$$

$$\phi = 1.1903 \text{ radian}$$

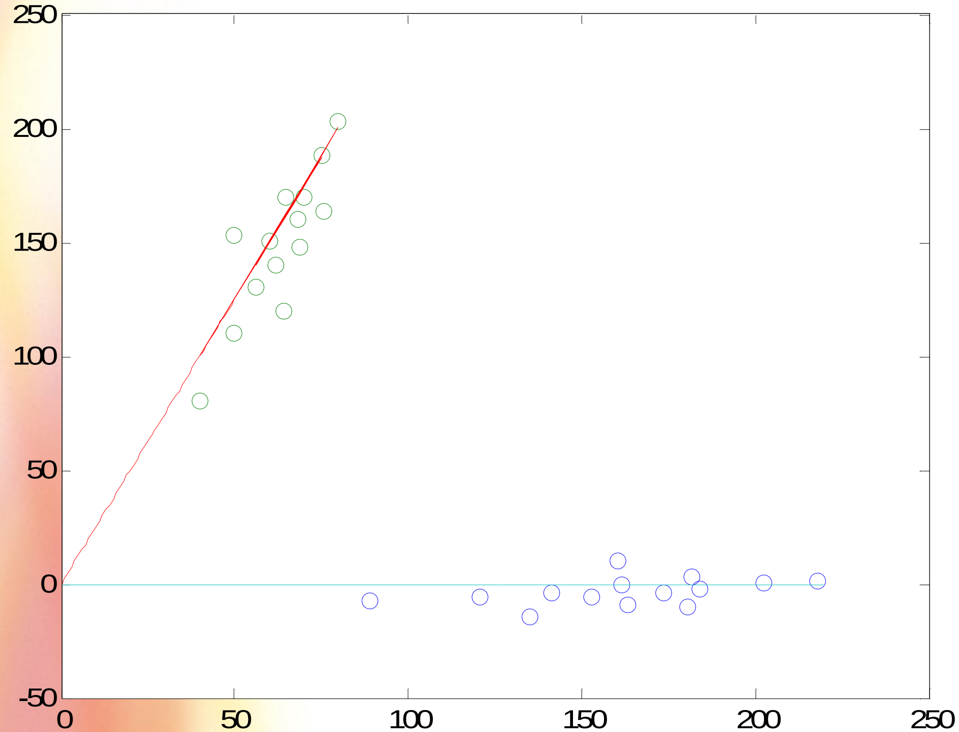
$$\cos\phi = 0.37139$$

$$\sin\phi = 0.92848$$

Exemple de motivation (3)

❑ Résultat de la rotation

- Toute l'énergie est concentrée autour la première coordonnée.



First Coordinate	Second Coordinate
182	3
202	0
162	0
184	−2
141	−4
218	1
174	−4
121	−6
90	−7
161	10
163	−9
153	−6
181	−9
135	−15

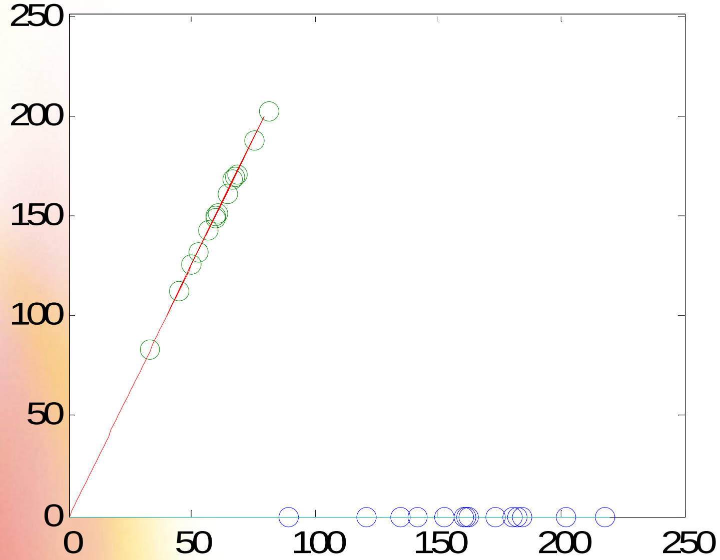
Exemple de motivation (4)

- Compression
  - On peut éliminer la seconde coordonnée en mettant toute les valeurs à zéro.
  - Ceci réduit de moitié les valeurs à encoder.
  - Mais aussi il y a un certaine distorsion qui résulte suite à cette compression.
- Reconstruction
  - On applique la transformation inverse (rotation dans le sens inverse).

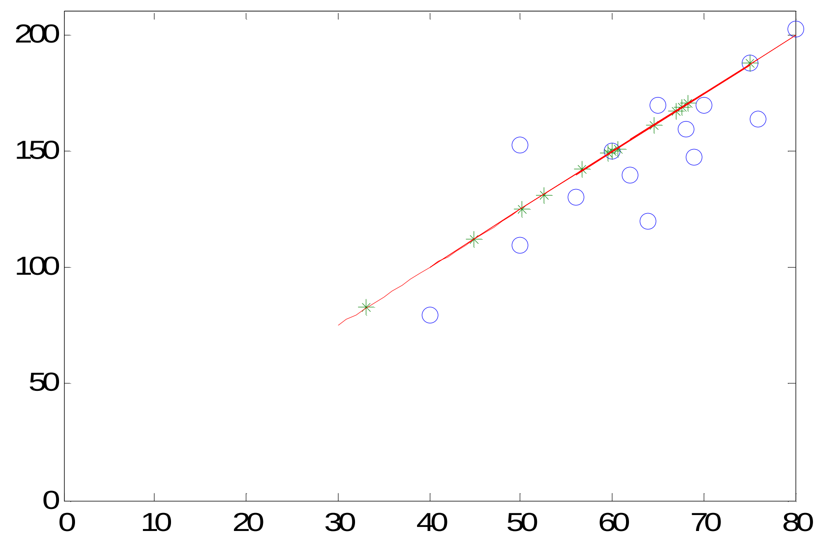
$$\text{Matrice de rotation inverse: } A^{-1} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

Exemple de motivation (5)

Rotation inverse



Comparaison entre les données originales et reconstruites



Exemple de motivation (6)

Données originales

Height	Weight
65	170
75	188
60	150
70	170
56	130
80	203
68	160
50	110
40	80
50	153
69	148
62	140
76	164
64	120

Données reconstruites

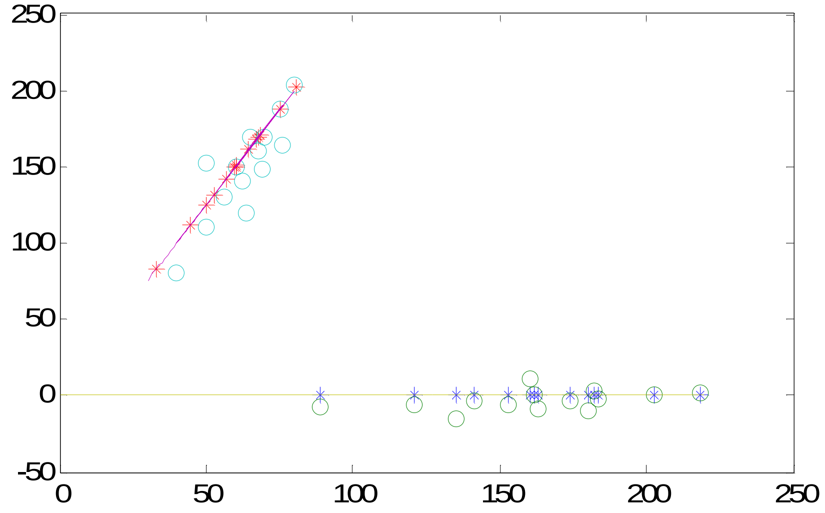
Height	Weight
68	169
75	188
60	150
68	171
53	131
81	203
65	162
45	112
34	84
60	150
61	151
57	142
67	168
50	125

Exemple de motivation (7)

Erreur de reconstruction

- L'erreur de reconstruction est la même que l'erreur induite par l'élimination de la seconde coordonnée de θ.

First Coordinate	Second Coordinate
182	3
202	0
162	0
184	−2
141	−4
218	1
174	−4
121	−6
90	−7
161	10
163	−9
153	−6
181	−9
135	−15



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

$$MSE = 3^2 + 2^2 + 4^2 + 1^2 + 4^2 + 6^2 + \ldots$$

$$MSE = 662.17$$

Karhuen-Loéve Transform (KLT) [1]

❑ Transformation Hotelling

- Cette transformation considère les vecteurs propres de la matrice d'autrocorrélation: $[R]_{i,j} = E[X_n X_{n+|i-j|}]$
- Cette transformation minimise l'erreur géométrique moyenne : $G = \sqrt[n]{x_1 x_2 \ldots x_n}$
- Elle assure le gain de codage par transformation maximal → transformation optimale.
- Cependant, cette transformation dépend des données.
- La transformation (la matrice d'autocorrélation) doit être envoyée au récepteur pour procéder à la reconstruction.
- Le gain de compression peut être supprimée lors de l'envoi de la transformation.
- La transformation est étudiée pour connaitre les limites.

Karhuen-Loéve Transform (KLT) [2]

- Exemple KLT de taille 2
  - Matrice d'autocorrélation de la source:

$$R = \begin{bmatrix} R_{xx}(0) & R_{xx}(1) \\ R_{xx}(1) & R_{xx}(0) \end{bmatrix}$$

  - Détermination des valeurs propres revient à résoudre cette équation: $\det(\lambda I - R) = 0$

$$\det\left(\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} R_{xx}(0) & R_{xx}(1) \\ R_{xx}(1) & R_{xx}(0) \end{bmatrix}\right) = 0$$

$$\det\left(\begin{bmatrix} \lambda - R_{xx}(0) & R_{xx}(1) \\ R_{xx}(1) & \lambda - R_{xx}(0) \end{bmatrix}\right) = 0$$

$$(\lambda - R_{xx}(0))^2 - (R_{xx}(1))^2 = 0$$

$$\lambda_1 - R_{xx}(0) = R_{xx}(1) \text{ et } \lambda_2 - R_{xx}(0) = -R_{xx}(1)$$

$$\lambda_1 = R_{xx}(0) + R_{xx}(1) \text{ et } \lambda_2 = R_{xx}(0) - R_{xx}(1)$$

Karhuen-Loéve Transform (KLT) [3]

Exemple KLT de taille 2 (suite)

- Détermination des vecteurs propres:

$$R V_1 = \lambda_1 V_1$$

$$R V_2 = \lambda_2 V_2$$

$$\lambda_1 = R_{xx}(0) + R_{xx}(1) \text{ et } \lambda_2 = R_{xx}(0) - R_{xx}(1)$$

$$\begin{bmatrix} R_{xx}(0) & R_{xx}(1) \\ R_{xx}(1) & R_{xx}(0) \end{bmatrix} \begin{bmatrix} V_1(0) \\ V_1(1) \end{bmatrix} = \lambda_1 \begin{bmatrix} V_1(0) \\ V_1(1) \end{bmatrix}$$

$$\begin{bmatrix} R_{xx}(0)V_1(0) + R_{xx}(1)V_1(1) \\ R_{xx}(1)V_1(0) + R_{xx}(0)V_1(1) \end{bmatrix} = \begin{bmatrix} R_{xx}(0)V_1(0) + R_{xx}(1)V_1(0) \\ R_{xx}(0)V_1(1) + R_{xx}(1)V_1(1) \end{bmatrix}$$

$$R_{xx}(1)V_1(1) = R_{xx}(1)V_1(0) \rightarrow V_1(0) = V_1(1) = \alpha$$

$$\text{de la même façon: } V_2(0) = -V_2(1) = \beta$$

$$\text{Vecteurs propres: } V_1 = \begin{bmatrix} \alpha \\ \alpha \end{bmatrix}, \quad V_2 = \begin{bmatrix} \beta \\ -\beta \end{bmatrix}$$

Mohamed Haj Taieb, Département de génie électrique et de génie informatique

Karhuen-Loéve Transform (KLT) [4]

- Exemple KLT de taille 2 (suite)
  - Supposons que les vecteurs propres sont orthonormales (on sait déjà que les vecteurs propres sont orthogonales):

$$\text{Vecteurs propres: } V_1 = \begin{bmatrix} \alpha \\ \alpha \end{bmatrix}, \quad V_2 = \begin{bmatrix} \beta \\ -\beta \end{bmatrix}$$

$$V_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \quad V_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$\text{Matrice de transformation KLT}$$

$$K = [V_1 \ V_2] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

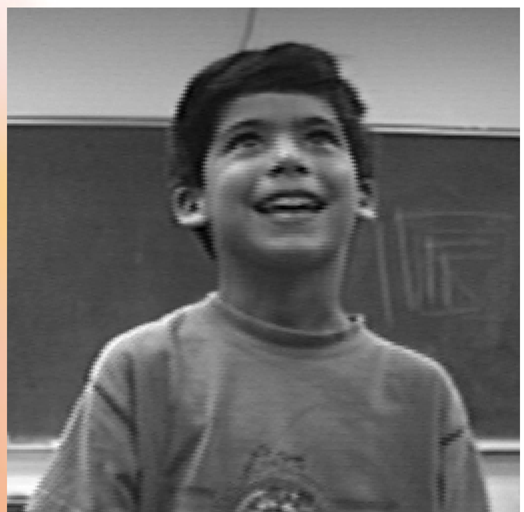
Karhuen-Loéve Transform (KLT) [5]

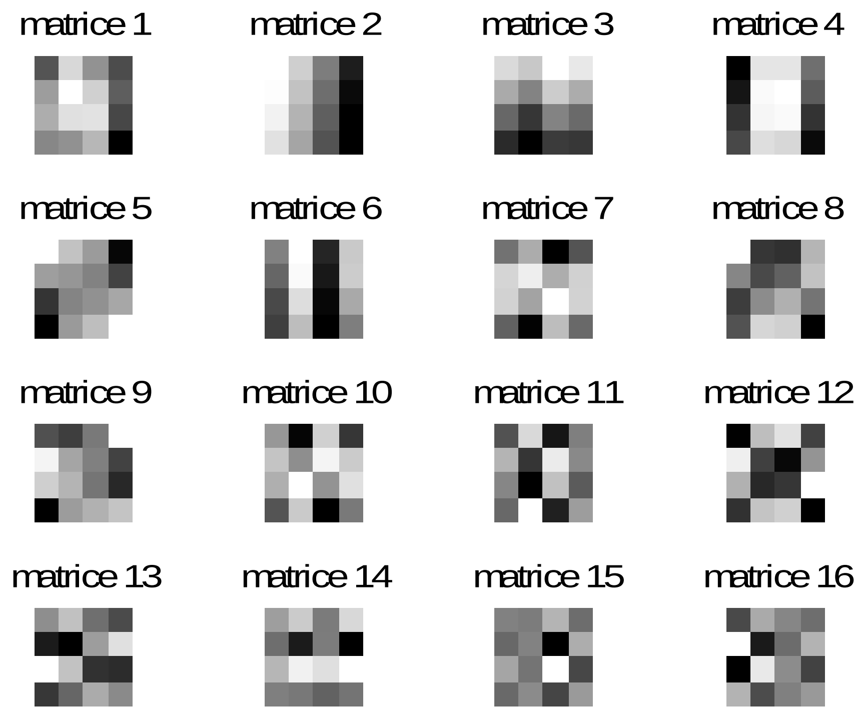
- Application pour la compression d'image
  - On découpe l'image en blocs de 4x4.
  - Chaque blocs, après redimensionnement, constitue un vecteur 16x1.
  - On détermine les vecteurs propres de ces matrices et on les ordonne suivant les valeurs propres décroissantes.
  - Ces vecteurs propres forme une matrice V 16x16.
- Transformation KLT
  - On redimensionne chaque bloc 4x4 en un vecteur X 16x1.
  - La transformation de ce vecteur: T=VX.
  - On remarque que l'énergie est concentrée autour des premiers coefficients.

Karhuen-Loéve Transform (KLT) [6]

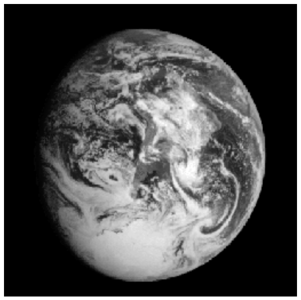
❑ Compression

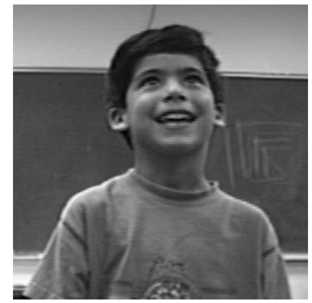
- On garde les premiers coefficients et on élimine les autres.
- Ou bien on quantifie avec plus de précision les premiers coefficients.

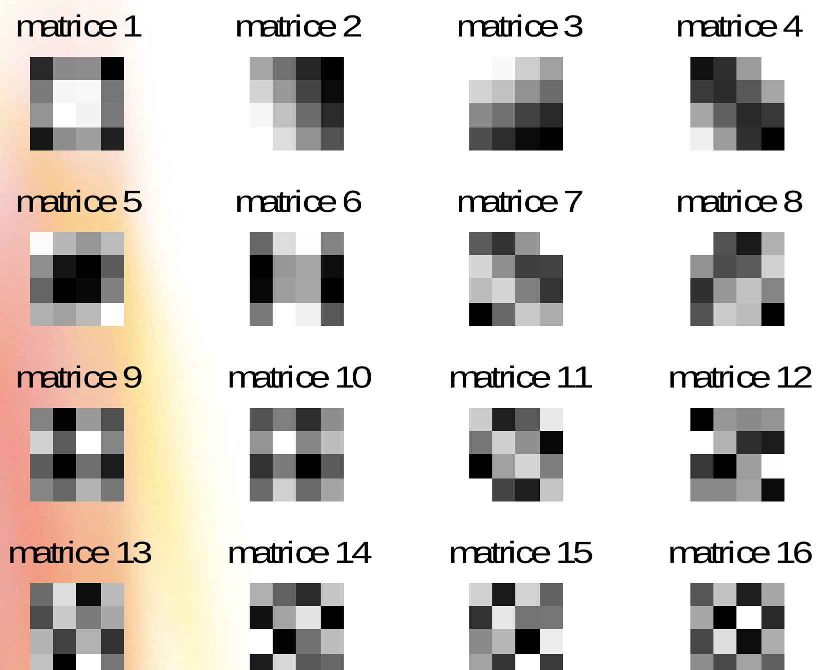


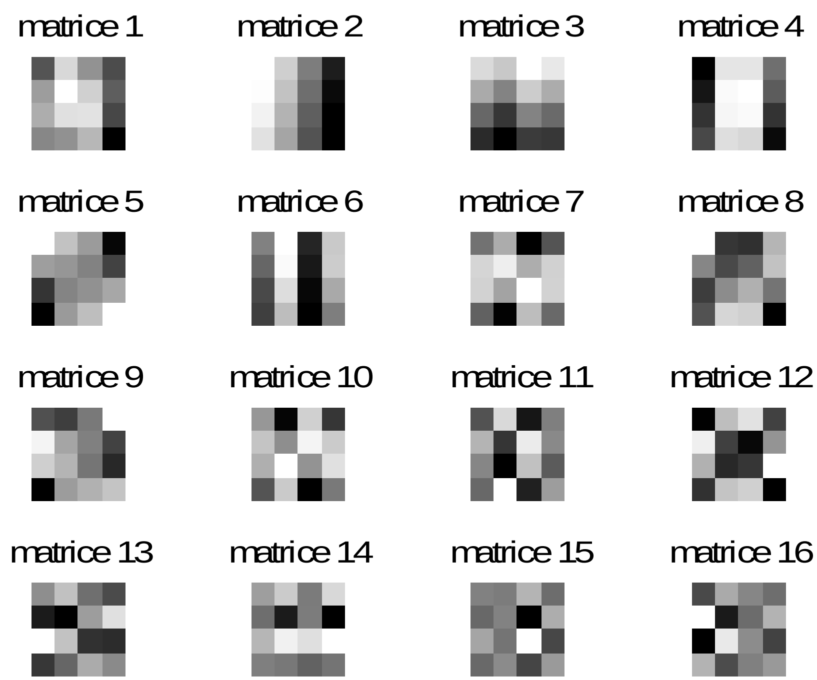


Karhuen-Loéve Transform (KLT) [7]









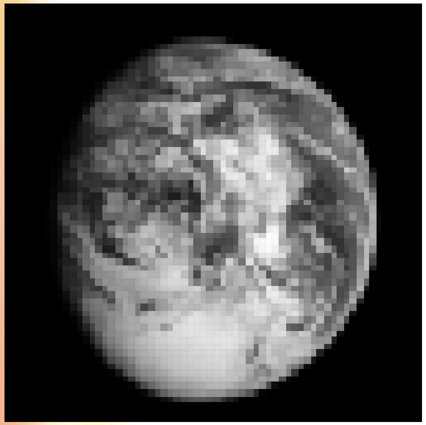
Karhuen-Loéve Transform (KLT) [8]

Élimination de 15 coefficients

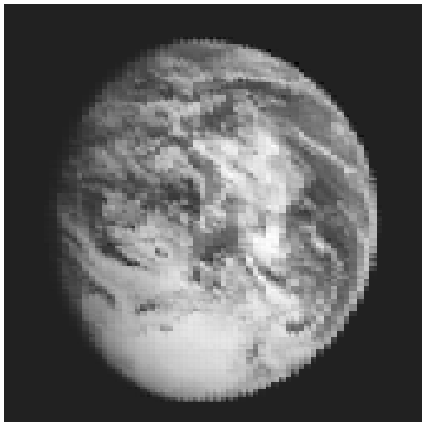
Élimination de 14 coefficients

Élimination de 13 coefficients

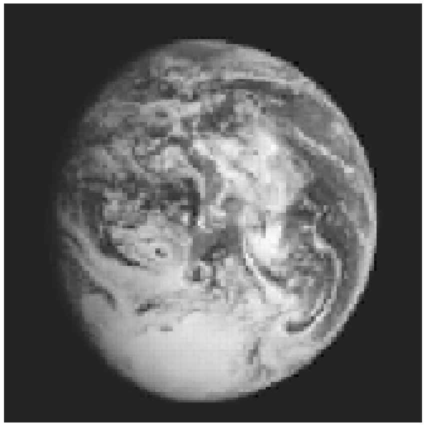
Élimination de 15 coefficients



Élimination de 14 coefficients



Élimination de 13 coefficients



Discrete Cosine Transform (DCT) [1]

DCT

- Les coefficients de la matrice C de la transformation DCT sont donnés par :

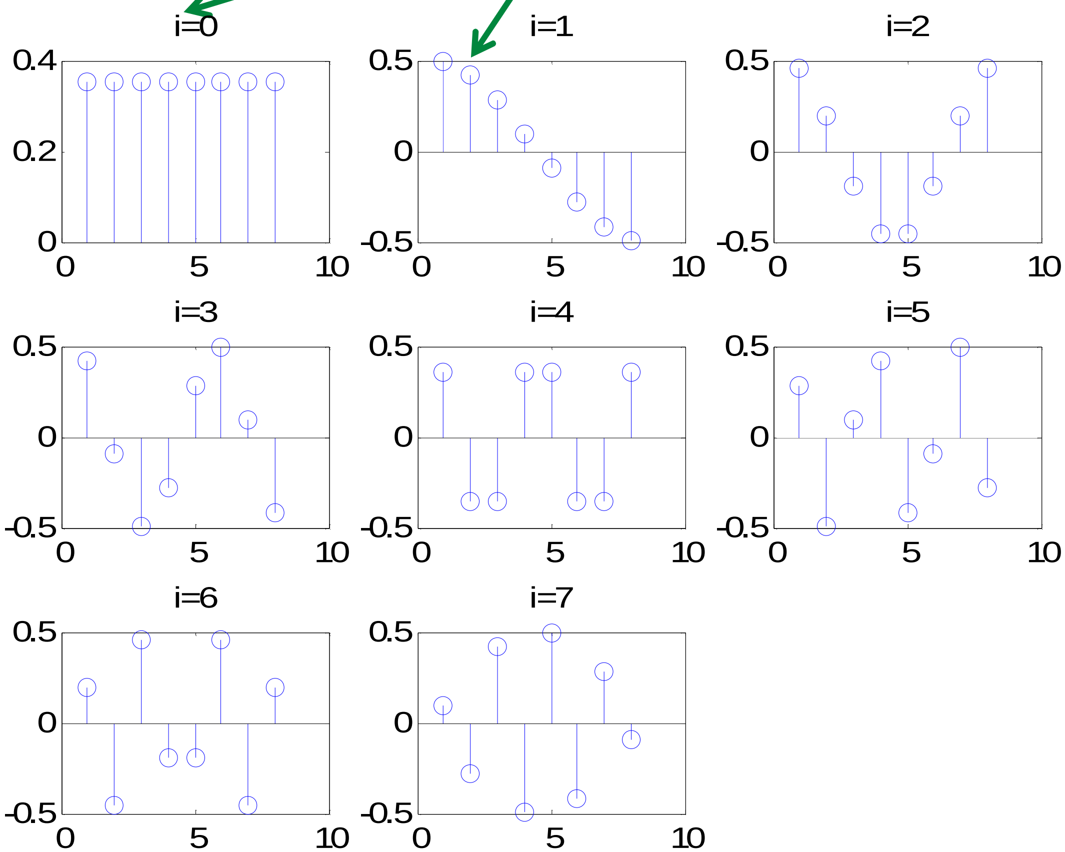
$$C_{i,j} = \begin{cases} \sqrt{\frac{1}{N}} \cos \frac{(2j+1)i\pi}{2N} & i=0, \quad j=0,1,\ldots,N-1 \\ \sqrt{\frac{2}{N}} \cos \frac{(2j+1)i\pi}{2N} & i=1,2,\ldots,N-1, \ j=0,1,\ldots,N-1 \end{cases}$$

- Contrairement à la KLT, la DCT est indépendante des données à encoder.
- Pas besoin de transmettre la base de la transformation au récepteur.
- La DCT est efficace à la compaction d'énergie lorsqu'il y a une corrélation significative entre les échantillons de la source. Exemple: pixels d'une image.

Discrete Cosine Transform (DCT) [2]

❑ Vecteurs de Bases de la DCT

$$C_{i,j} = \begin{cases} \sqrt{\dfrac{1}{N}} \cos \dfrac{(2j+1)i\pi}{2N} & i = 0, \quad j = 0,1,\ldots,N-1 \\ \sqrt{\dfrac{2}{N}} \cos \dfrac{(2j+1)i\pi}{2N} & i = 1,2,\ldots,N-1, \ j = 0,1,\ldots,N-1 \end{cases}$$



Discrete Cosine Transform (DCT) [3]

DCT

- Le produit tensoriel des lignes de la matrice C

$$u \otimes v = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \otimes \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}$$

$$u \otimes v = \begin{bmatrix} u_1v_1 & u_1v_2 & u_1v_3 \\ u_2v_1 & u_2v_2 & u_2v_3 \\ u_3v_1 & u_3v_2 & u_3v_3 \end{bmatrix}$$