

Autres bases orthogonales

- Transformée en cosinus discrète (TCD, en anglais discrete cosine transform: DCT)
- TCD locale
- Transformée à fenêtre coulissante (Gabor et autres)
- Analyse en composante principal

Transformée de Fourier en cosinus

La transformée de Fourier en cosinus d'une fonction $g(t)$ est la transformée de Fourier de la partie paire de $g(t)$. Si $g_{\text{pair}}(t) = g_p(t)$:

$g_p(t) = g(t)$ si $t > 0$; et $g_p(t) = g(-t)$ si $t < 0$.

Par définition $TF[g(t)] = \int_{-\infty}^{\infty} g(t) e^{-i2\pi f t} dt$

$$TF[g(t)] = \int_{-\infty}^{\infty} g(t) e^{-i2\pi f t} dt = \int_{-\infty}^{\infty} g_p(t) [\cos(2\pi f t) - i \sin(2\pi f t)] dt$$

$$G_{\cos}(f) = G_c(f) = 2 \int_0^{\infty} g(t) \cos(2\pi f t) dt$$

Ex. Trouver la transformée de Fourier en cosinus de $g(t) = \exp(-a|t|)$ avec $a > 0$ pour $|t| < \infty$.

Réponse : $G(f) = 2a / (a^2 + 4\pi^2 f^2)$

Transformée de Fourier en sinus

La transformée de Fourier en sinus d'une fonction $g(t)$ est la transformée de Fourier de la partie impaire de $g(t)$. Si $g_{\text{impair}}(t) = g_i(t)$:

$g_i(t) = g(t)$ si $t > 0$; et $g_i(t) = -g(-t)$ si $t < 0$.

Par définition $TF[g(t)] = \int_{-\infty}^{\infty} g(t) e^{-i2\pi f t} dt$

$$TF[g(t)] = \int_{-\infty}^{\infty} g(t) e^{-i2\pi f t} dt = \int_{-\infty}^{\infty} g_i(t) [\cos(2\pi f t) - i \sin(2\pi f t)] dt$$

$$G_{\sin}(f) = G_s(f) = -2i \int_0^{\infty} g(t) \sin(2\pi f t) dt$$

La transformée de Fourier en cosinus discrets (DCT dans la Toolbox image processing de Matlab)

- Extrêmement important pour les signaux **réels** (sons, images, vidéos)
- Possibilité d'exploiter la propriété de symétrie (TF d'une fonction réelle et symétrique est réelle et symétrique)
- Toute l'information du signal est concentrée dans les basses fréquences. C'est la propriété de "strong energy compaction". C'est pourquoi la DCT est à la base de JPEG, MPEG. Très bon pour la compression.
- Moyen de faire mieux en faisant de la DCT locale. C'est-à-dire faire N approximations locales en faisant N DCT indépendante dans des fenêtres locales. (*locDCT*)

Voir diapositives, notes, TP 4 et démonstrations faites en classe.

Les limites de la Transformée de Fourier

Défauts majeurs de la TF

- 1- Une information sur un moment du signal est répartie parmi toutes fréquences de sa transformée (pensez à la TF[Dirac] = 1).
- 2- Pas d'information sur le temps ou espace. La TF donne le contenu fréquentiel d'un signal sans donner la localisation. Il est possible de savoir quand ou où une certaine a eu lieu dans le signal original. Ce manque d'information sur le temps (espace) rend une TF terriblement sensible aux erreurs.

Exemples :

- Si nous enregistrons un signal d'une heure et que les 5 dernières minutes sont corrompues, alors cette erreur corrompt TOUTE la TF.
- Les erreurs de phases sont désastreuses. Elles risquent d'engendrer un signal totalement différent du signal original

Transformée de Karhunen-Loeve (TKL)

La TKL ou APC (PCA en anglais et dans la Image Processing Toolbox de Matlab) est une représentation extrêmement utilisée en analyse de données et d'images. Elle est aussi appelée Transformée de Hotelling, Transformée en composantes principales, Transformée en vecteurs propres.

Cette section est pour votre culture personnelle et pour vos connaissances en tant qu'expert en imagerie. Vous ne serez pas questionné sur cette partie dans l'examen final.

La transformation des coordonnées: rotation

La rotation d'un repère d'un angle θ est équivalente à une transformation des coordonnées par une matrice $\begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$. Ainsi le point P de coordonnées (x,y) dans le repère (x,y) a pour coordonnées (X,Y) dans le nouveau repère (X,Y) obtenu par rotation de (x,y) par un angle θ (Figure ci-dessous).

Ceci peut être démontré comme suit et selon la figure ci-dessous:

$$X = r \cos(\phi)$$

$$Y = r \sin(\phi)$$

et

$$x = r \cos(\phi + \theta) = r(\cos(\phi)\cos(\theta) - \sin(\phi)\sin(\theta)) = [r \cos(\phi)]\cos(\theta) - [r \sin(\phi)]\sin(\theta) = [X]\cos(\theta) - [Y]\sin(\theta)$$

$$y = r \sin(\phi + \theta) = r(\sin(\phi)\cos(\theta) + \cos(\phi)\sin(\theta)) = [r \sin(\phi)]\cos(\theta) + [r \cos(\phi)]\sin(\theta) = [Y]\cos(\theta) + [X]\sin(\theta)$$

qui s'écrit comme suit:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \Rightarrow \begin{cases} x = X \cos(\theta) - Y \sin(\theta) \\ y = X \sin(\theta) + Y \cos(\theta) \end{cases}$$

Pour chercher l'expression de (X,Y) en fonction de (x,y), prendre le système d'équations précédent et le résoudre pour (X,Y):

$$x = X \cos(\theta) - Y \sin(\theta)$$

$$y = X \sin(\theta) + Y \cos(\theta)$$

De la 1ère équation: $X = \frac{x + Y \sin(\theta)}{\cos(\theta)}$ à substituer dans la 2ème équation:

$$y = \frac{x + Y \sin(\theta)}{\cos(\theta)} \sin(\theta) + Y \cos(\theta) = \frac{x \sin(\theta) + Y \sin^2(\theta) + Y \cos^2(\theta)}{\cos(\theta)} = \frac{x \sin(\theta) + Y}{\cos(\theta)}$$

$$\Rightarrow Y = -x \sin(\theta) + y \cos(\theta) \Rightarrow$$

$$X = \frac{x + Y \sin(\theta)}{\cos(\theta)} = \frac{x + (-x \sin(\theta) + y \cos(\theta)) \sin(\theta)}{\cos(\theta)} = \frac{x(1 - \sin^2(\theta)) + y \cos(\theta) \sin(\theta)}{\cos(\theta)} =$$

$$\frac{x \cos^2(\theta) + y \cos(\theta) \sin(\theta)}{\cos(\theta)} = x \cos(\theta) + y \sin(\theta)$$

$$\Rightarrow \begin{cases} X = x \cos(\theta) + y \sin(\theta) \\ Y = -x \sin(\theta) + y \cos(\theta) \end{cases}$$

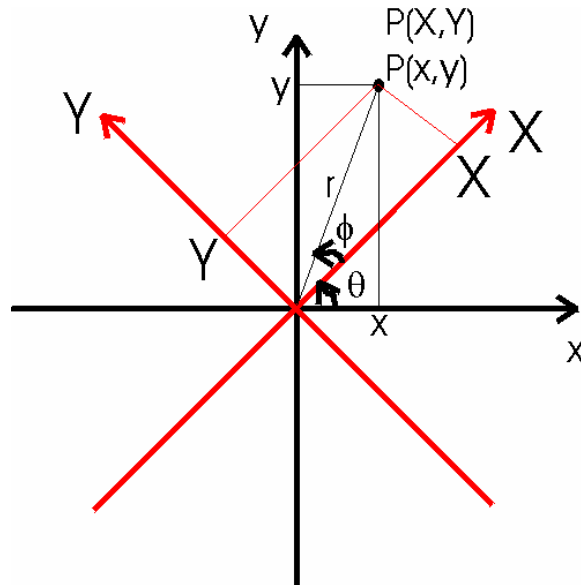


Figure. Rotation du repère xy vers le repère XY par un angle de rotation θ .

Compression de données

TKL permet de décorrélérer un signal et de compacter les informations contenues dans un signal.

Dans une image, il y a une grande corrélation entre les pixels voisins. C'est à dire, en connaissant l'intensité d'un pixel, on peut savoir l'intensité de son voisin.

Exemple 1.

Le poids est corrélé à la taille (ici les taille 0 cm et poids 0 kg sont mis pour forcer le passage de la régression linéaire par 0 pour faciliter le calcul de la rotation).

Taille (cm)	Poids (kg)
a=[0	0
162.5	76.5
187.5	84.6
150	67.5
175	76.5
140	58.5

200	91.35
170	72
125	49.5
100	36];

Mettre $a(:,1)$ = taille; $a(:,2)$ = poids. Ajuster les données par une droite et l'afficher:
`p=polyfit(a(:,1),a(:,2),1); y=polyval(p,a(:,1));figure;plot(a(:,1),a(:,2),'bo',a(:,1),y,'r-','linewidth',2);set(gca,'xcolor',[0 0 0],'FontSize',16);set(gcf,'Color',[1 1 1],'Position',[280 100 800 600],'InvertHardCopy','off'); xlabel('Taille (cm)');ylabel('Poids (kg)');print -dtiff tkl-poids-taille1`

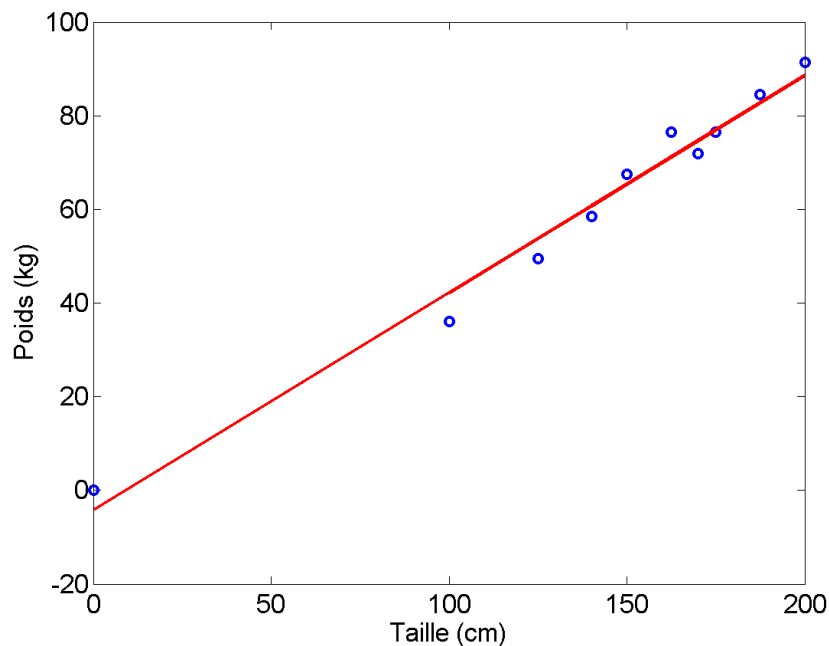


Figure. TKL-poids=f(taille).

Trouver l'angle que fait la droite avec l'axe des x. La pente étant la tangente de l'angle cherché:

$\theta = \text{atand}(p(1)) = 24.918^\circ$; % $p(2)$ étant le coefficient de x^0 selon $y = p(1)x^1 + p(2)x^0$ ou bien $y = ax+b$.

En appliquant la rotation avec un angle 24.918° à la matrice a:

`ar = (R * a)'`; % l'apostrophe indique la transposée.

et $R = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} = [\text{cosd}(\theta) \text{ sind}(\theta); -\text{sind}(\theta) \text{ cosd}(\theta)]$;

ar = [0	0
179.604191721671	0.91340900220375	
205.689669545723	-2.27372869342308	
164.475897252689	-1.9822267586915	
190.940566996188	-4.35314543455311	
151.614877838611	-5.93117340693813	
219.869984216088	-1.41864048194086	
184.510057289148	-6.32761875867642	
134.219308314726	-7.773498280482	
105.858678973994	-9.48367470344645];	

```
figure;plot(ar(:,1),ar(:,2),'bo','linewidth',2);set(gca,'xcolor',[0 0
0],'FontSize',16);set(gcf,'Color',[1 1 1],'Position',[280 100 800 600],'InvertHardCopy','off');
xlabel('Taille (cm)');ylabel('Poids (kg)');print -dtiff tkl-poids-taille2
```

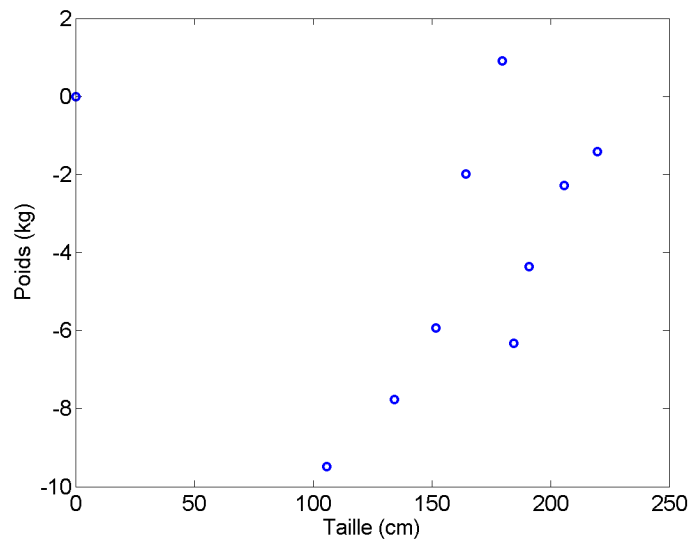


Figure. Affichage de ar.

En mettant tous les poids à la moyenne de -4 kg ce qui permet de compresser les données puisqu'on ne garde que les valeurs de la taille et une seule valeur pour le poids.

Pour retrouver les valeurs réelles des tailles et des poids, faire la transformée inverse:

```
ar =[ 0 -4
179.604191721671 -4
205.689669545723 -4
164.475897252689 -4
190.940566996188 -4
151.614877838611 -4
219.869984216088 -4
184.510057289148 -4
134.219308314726 -4
105.858678973994 -4];
```

$$Rr = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} = [\cosd(\theta) \ -sind(\theta); \ sind(\theta) \ \cosd(\theta)]$$

arr=(Rr*ar)'; % Alternativement on utilise l'inverse de R au lieu de Rr: arri=(inv(R)*ar)';

```
arr =[ 1.6852974197622 -3.62764008784539
164.570138878413 72.0439801339063
188.227320144696 83.0344272514411
150.850137009311 65.6700612254119
174.851211227587 76.8202711338061
139.186349610043 60.2514005168974
```

201.087589633816	89.0089391827868
169.019317527953	74.1109407795489
123.410133271107	52.922223408423
97.6895942928666	40.9731995457315];

Afficher a et arr pour comparaison:

```
figure;plot(a(:,1),a(:,2),'bo',
arr(:,1),arr(:,2),'ro','linewidth',2);legend({'a','arr'});set(gca,'xcolor',[0 0
0],'FontSize',16);set(gcf,'Color',[1 1 1],'Position',[280 100 800 600],'InvertHardCopy','off');
xlabel('Taille (cm)');ylabel('Poids (kg)');print -dtiff tkl-poids-taille3
```

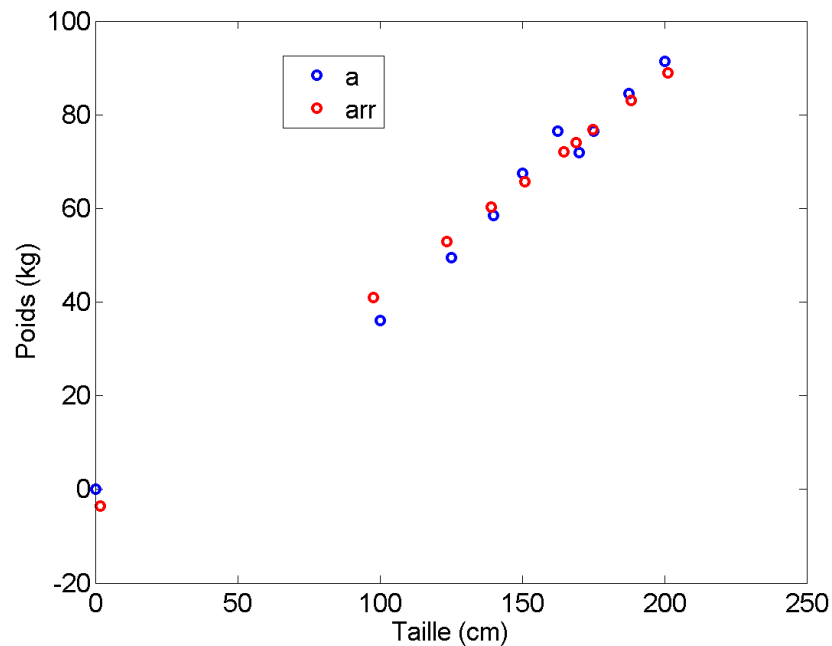


Figure. Matrice de départ a et matrice compressée puis reconstruite arr.

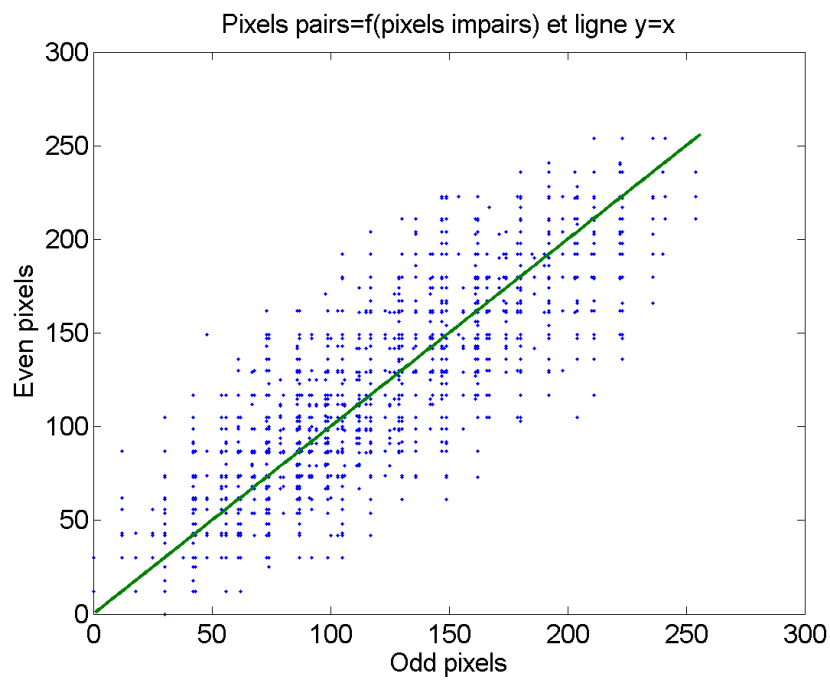
Exemple 2.

Prendre l'image de la figure suivante:

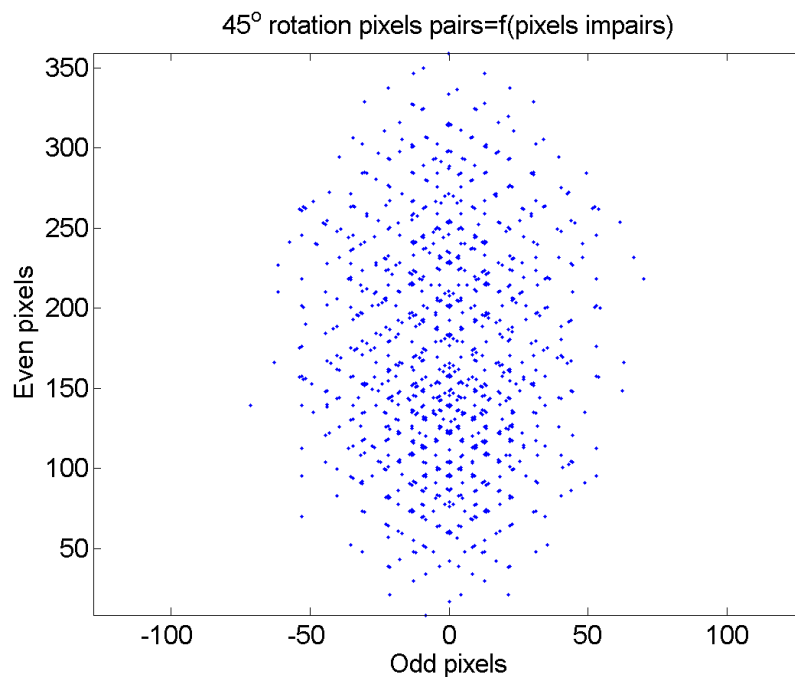
Image originale



Sélectionner les indices pairs des pixels sur les colonnes et les afficher en fonction des indices impairs à partir de l'image précédente, tracer une ligne de régression $y=x$.



Faire une rotation de 45° :



Procédure de calculs de la transformation:

- 1- Centrer les valeurs des pixels pairs = $f(\text{impairs})$ formant la matrice X , c'est-à-dire mettre leur moyenne 0 en retranchant la moyenne de leur intensité: $n = \text{size}(X,1); X0 = X - \text{ones}(n,1) * \text{mean}(X);$
- 2- Calculer la covariance: $c = X0' * X0 / (n-1);$
- 3- Calculer les vecteurs propres et les valeurs propres de la matrice de covariance. Les vecteurs propres forment une base orthogonale: $[v,d]=\text{eig}(c);$
- 4- Prendre la diagonale des valeurs propres et les normaliser de façon à obtenir leur total égal à 1: $dd=\text{diag}(d); dd=dd/\text{sum}(dd);$
- 5- Choisir les valeurs les plus importantes des valeurs propres normalisées. Ces valeurs choisies indiquent le nombre de composantes à retenir ainsi que leur position.
- 6- On peut vérifier l'orthonormalité des vecteurs propres en observant que la diagonale de la matrice suivante vaut 1: $w=v*v';$
- 7- Sélectionner les n_p vecteurs propres correspondants aux valeurs propres les plus importantes: $vs=v(:,\text{end}-n_p+1:\text{end});$
- 8- Calculer les composantes: $Y=(vs'*X0)';$
- 9- Finalement, reconstruire l'image:
 $yr=\text{reshape}(Y,128,256); \text{figuredb}; \text{colormap}(\text{map}); \text{imagesc}(yr);$



Figures. Différence entre Yr l'image compressée en éliminant la corrélation entre 2 pixels voisins, et l'image Xr ou les pixels voisins ont été directement éliminés.

Exemple 3.

Dans cet exemple, il ne s'agit pas de compresser des images, mais de mettre en évidence des structures qui ne sont pas apparentes sur les images de départ. L'algorithme décompose la séquence dynamique en plusieurs composantes ayant des comportements dans le temps différents, c'est-à-dire en composantes orthogonales. Ici la série d'images a été décomposée en 2 composantes: sang et tissu.

Les images sont obtenues avec l'imagerie par la tomographie d'émission par positrons (TEP). Le but est de mesurer le métabolisme du glucose dans le coeur d'un rat. Le glucose marqué par un émetteur de radioactivité (radiotraceur) est injecté dans le rat. La mesure de la concentration du radiotraceur dans le coeur du rat se fait en fonction du temps: 12 x 5 sec; 8 x 30 sec et 3 x 300 sec.

- L'image de tous les cadres montre l'ensemble des images où l'on constate l'accumulation du radiotraceur dans le coeur à mesure que le temps passe.
- La décomposition permet de ressortir des structures, qui, autrement, n'étaient pas apparente.
- Observer comment la décomposition a permis de faire apparaître les poumons (Composante image # 1) comme dans l'image anatomique à travers le thorax.

