

Démarche

Fonction *decouper(coin_x, coin_y, largeur, hauteur, seuil)*

Traite un timbre (une *patch*) de taille 0..toute l'image :

- 1 Si la *patch* est de taille < 1

Alors ne rien faire

- 2 Sinon

Mesurer l'écart type des 3 couleurs de cette patch

Si cette mesure $< seuil$

Alors donner la couleur moyenne de la patch à tous ses pixels

Sinon

Couper la patch en 4

Recommencer (appel récursif) avec chaque quart

Démarche (suite)

Fonction *mesurer_ecart_type(coin_x, coin_y, largeur, hauteur)*

Calcule σ des couleurs d'un timbre (une *patch*) de taille > 0 :

- 1 Calculer la moyenne de chaque couleur des pixels de cette *patch*
- 2 Pour chacune des 3 couleurs (r,v,b) :
calculer σ^2 des valeurs de la couleur
→ la variance σ^2 est déjà pondérée !

$$3 \text{ écart_type_de_patch} = \frac{\sqrt{\hat{\sigma}_r^2} + \sqrt{\hat{\sigma}_v^2} + \sqrt{\hat{\sigma}_b^2}}{3}$$

👉 La moyenne d'une couleur C dans une *patch* :

$$\frac{\text{somme des valeurs de } C \text{ dans la patch}}{\text{superficie de la patch}}$$

Démarche (suite)

Utilitaires :

- 1 Ouvrir le fichier d'image (couleur), charger (les pixels de) l'image et visualiser

```
from PIL import Image
from math import sqrt
#Ouverture du fichier d'image : si "png", il faut convertir (rien si "bmp")
im = Image.open("steve.png").convert("RGB")
mat_pixels = im.load() # charger en mémoire les pixels de l'image
w,h=im.size # soit w et h, respectivement, la largeur et la hauteur de l'image

im.show() # Voir l'image à l'écran
```

- ☞ mat_pixels sera une matrice $[i,j]$
- ☞ Chaque pixel de l'image : $r, g, b = mat_pixels[i,j]$
- ☞ Pour ré-écrire un pixel : $mat_pixels[i,j] = int(\bar{r}), int(\bar{v}), int(\bar{b})$

- 2 Découper l'image en quatre par la fonction (récursive) *decouper* :
 $decouper(0, 0, w, h, 5)$ $seuil = 5$ mais faites varier $\in 3..20$?

Démarche (suite)

Une image originale et sa conversion :

