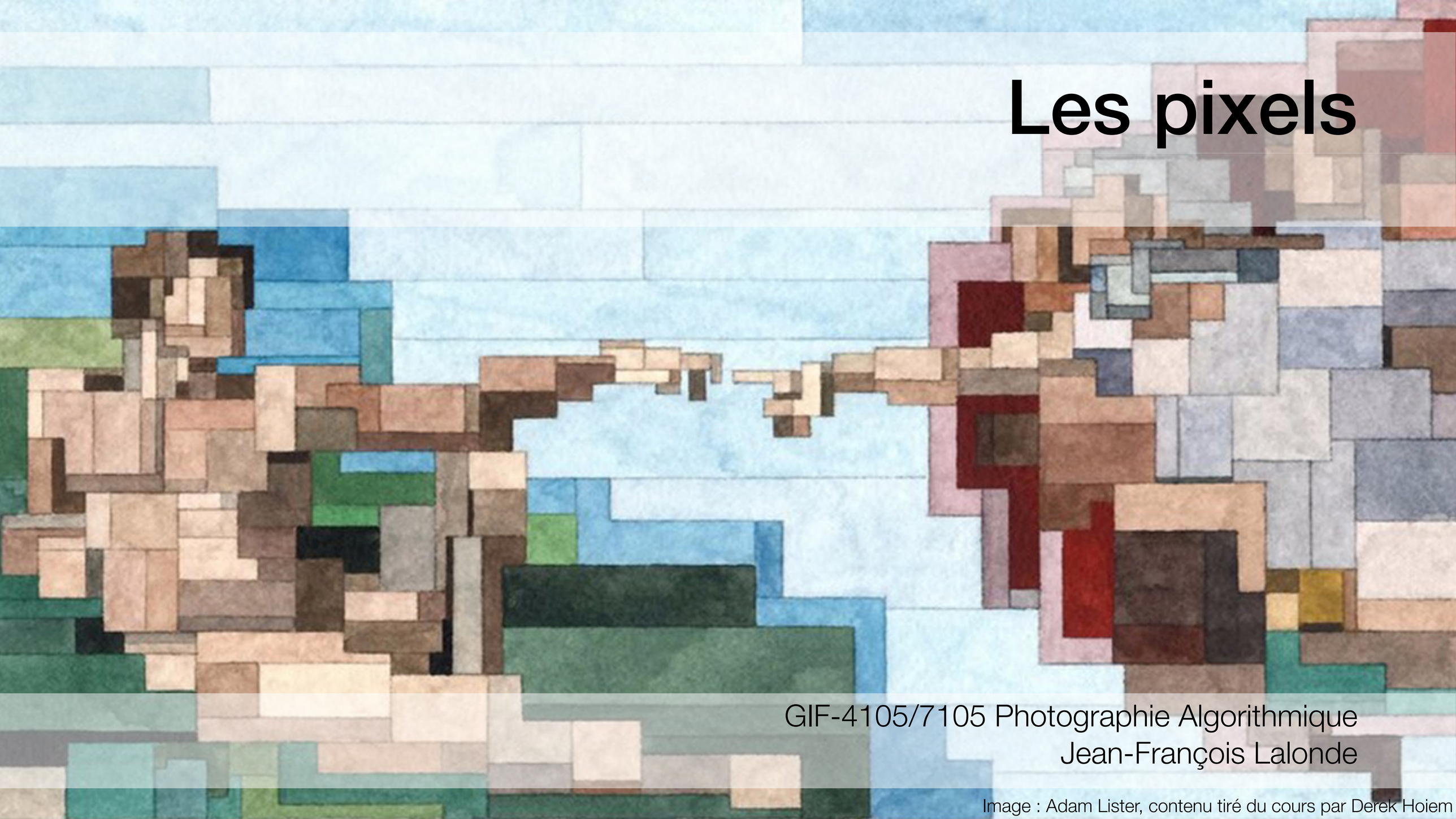
Les pixels



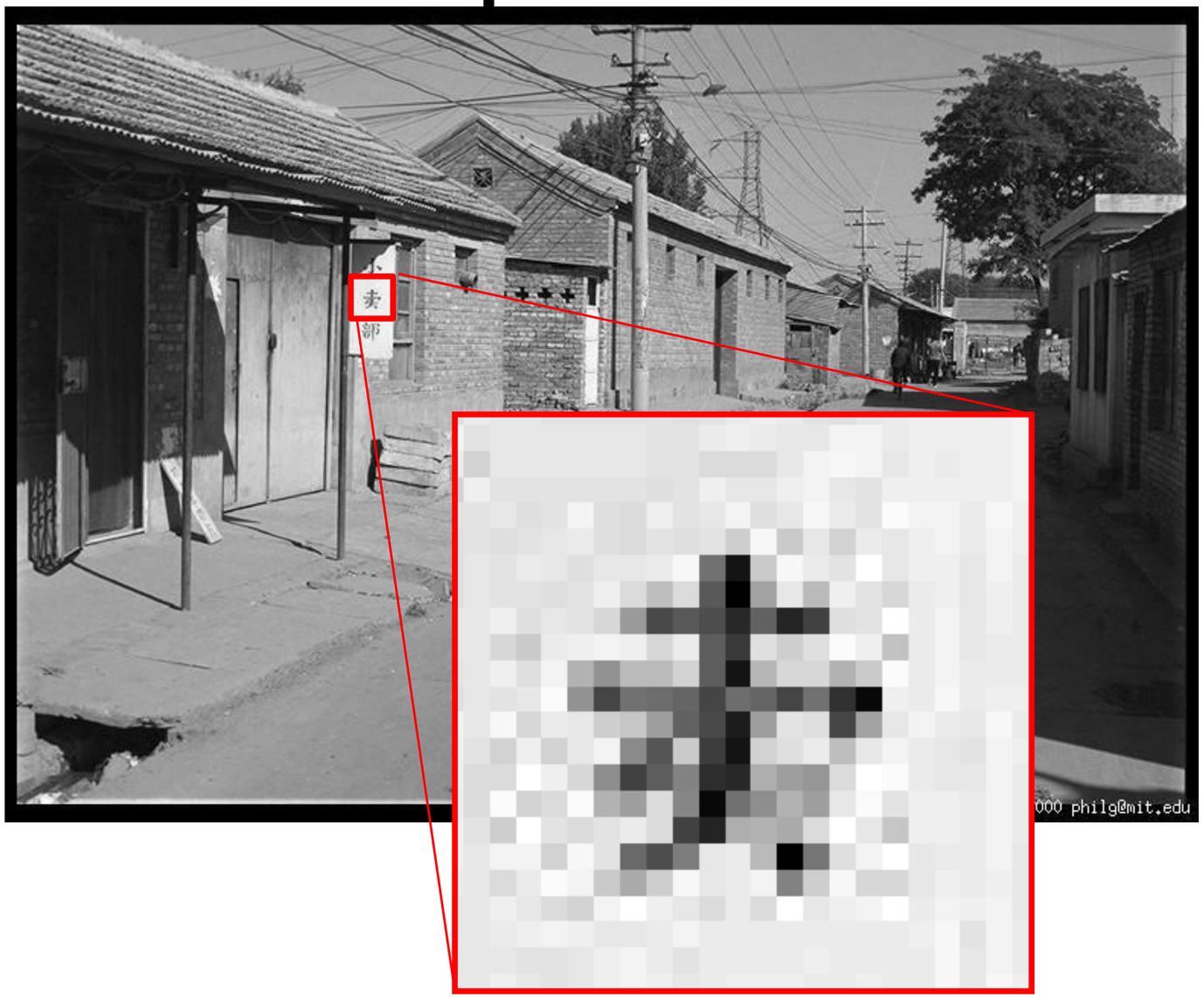
GIF-4105/7105 Photographie Algorithmique

Jean-François Lalonde

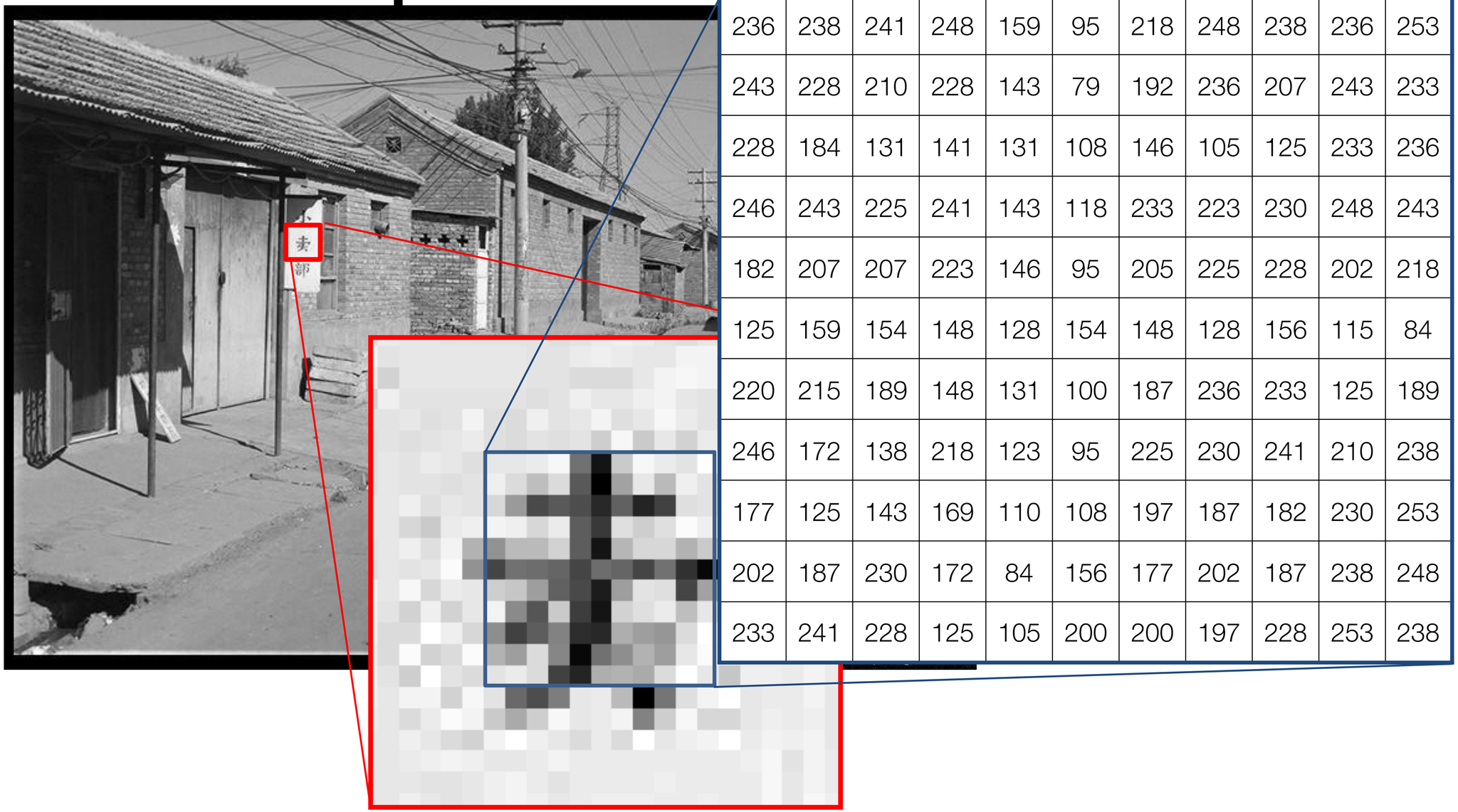
Image : Adam Lister, contenu tiré du cours par Derek Hoiem

Une matrice de pixels

Une matrice de pixels



Une matrice de pixels



236	238	241	248	159	95	218	248	238	236	253
243	228	210	228	143	79	192	236	207	243	233
228	184	131	141	131	108	146	105	125	233	236
246	243	225	241	143	118	233	223	230	248	243
182	207	207	223	146	95	205	225	228	202	218
125	159	154	148	128	154	148	128	156	115	84
220	215	189	148	131	100	187	236	233	125	189
246	172	138	218	123	95	225	230	241	210	238
177	125	143	169	110	108	197	187	182	230	253
202	187	230	172	84	156	177	202	187	238	248
233	241	228	125	105	200	200	197	228	253	238

Images dans python (gris)

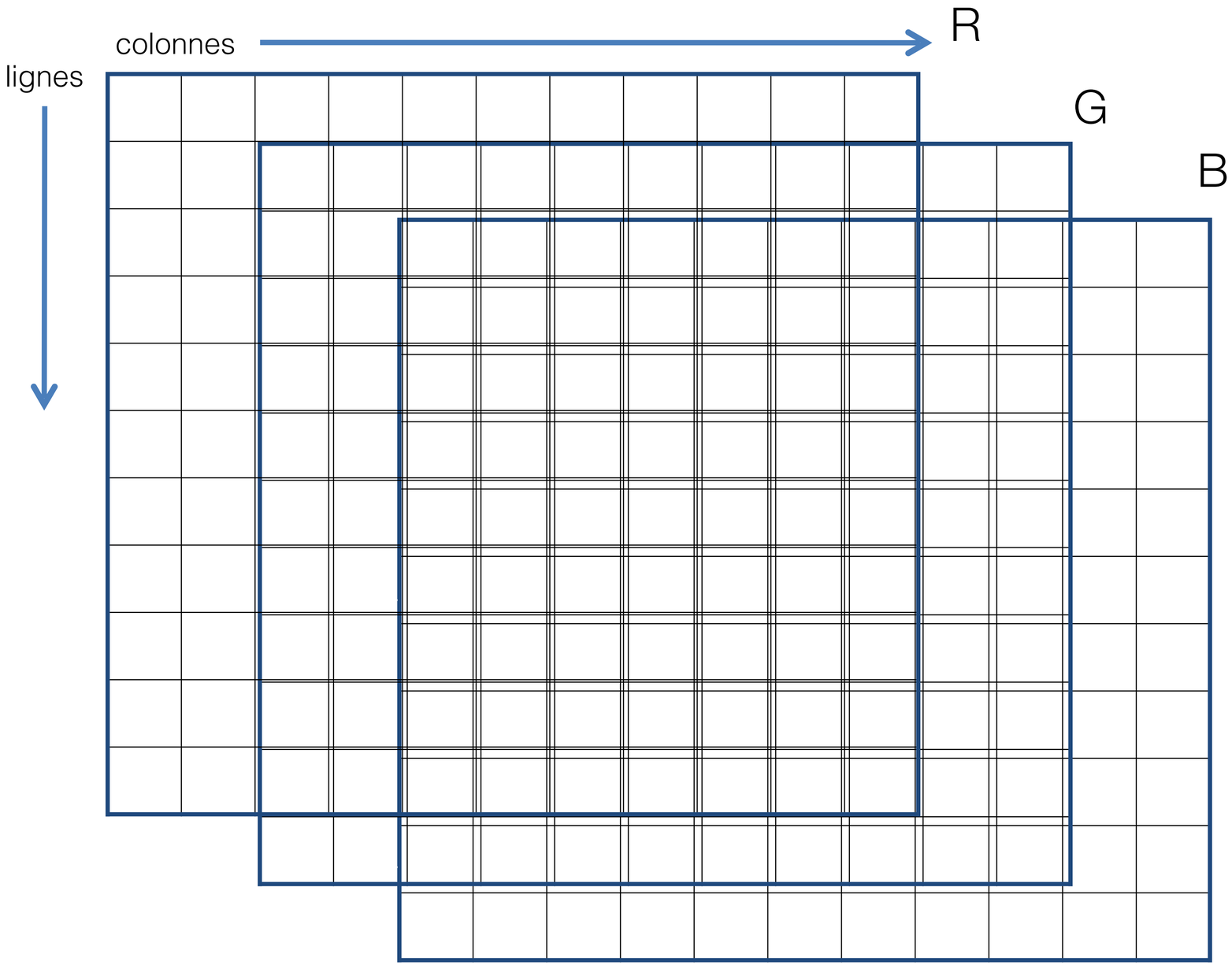
```
from skimage import io
img = io.imread('image.jpg')
```

- L'image est une matrice (numpy array) 2-D de dimensions NxM appelée `img`
  - `img[0,0]` : pixel en haut à gauche
  - `img[-1,-1]` : pixel en bas à droite
- `io.imread(filename)` retourne une image en « uint8 »
  - Chaque pixel contient une valeur non-signée encodée sur 8 bits, donc de 0 à 255, représentant l'intensité du pixel

Images dans python (couleur)

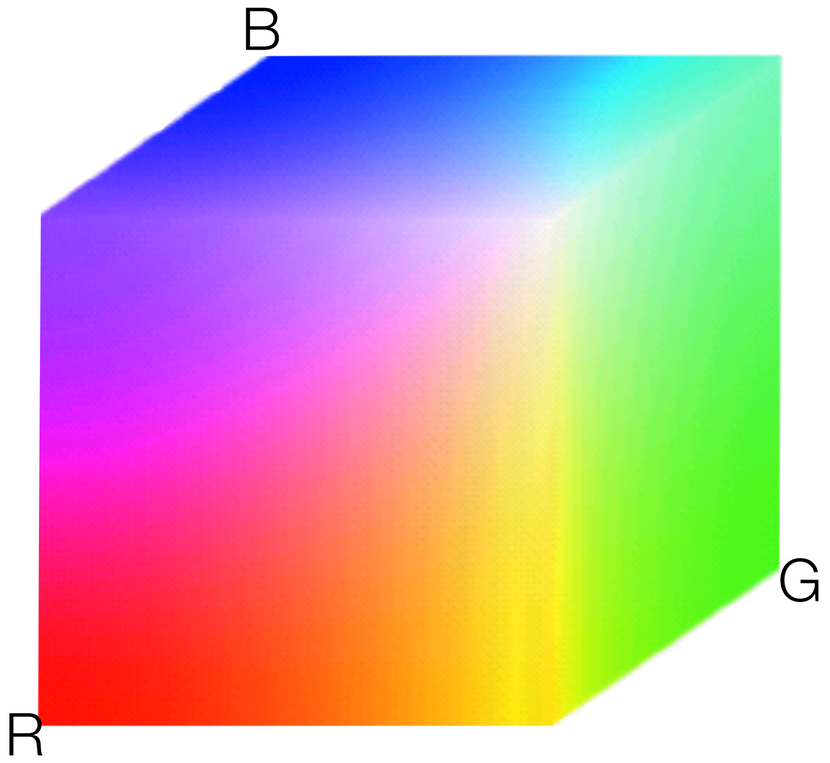
```
from skimage import io
img = io.imread('image.jpg')
```

- L'image est une matrice (numpy array) 3-D de dimensions NxMx3 appelée `img`
  - `img[0,0,0]` : pixel en haut à gauche, canal « R » (rouge)
  - `img[-1,-1,2]` : pixel en bas à droite, canal « B » (bleu)
- `io.imread(filename)` retourne une image en « uint8 »
  - Chaque pixel contient une valeur non-signée encodée sur 8 bits, donc de 0 à 255, représentant l'intensité du pixel


colonnes
lignes
R
G
B

Espace de couleur : RGB

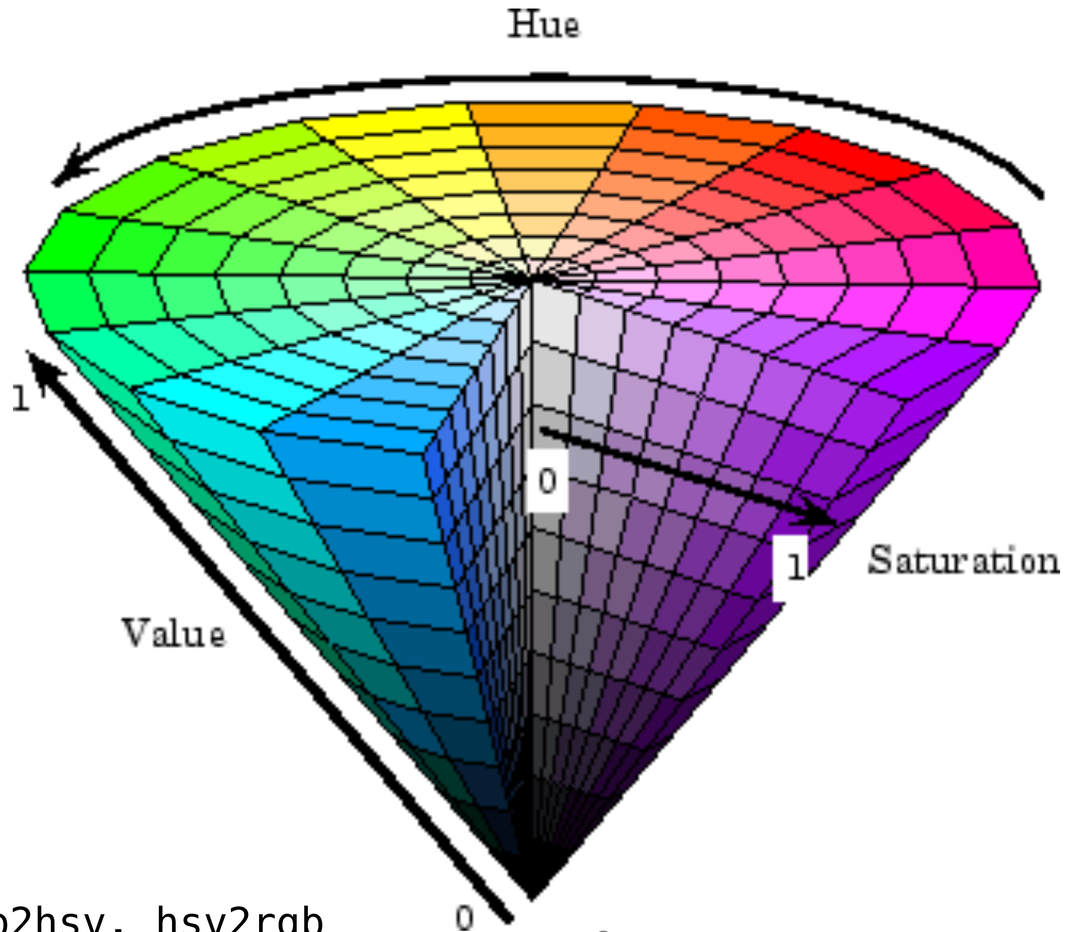
- Cube RGB
  - Pratique pour les appareils
  - Pas un modèle perceptuel
  - Où sont les gris?
  - Où sont la saturation et la teinte?

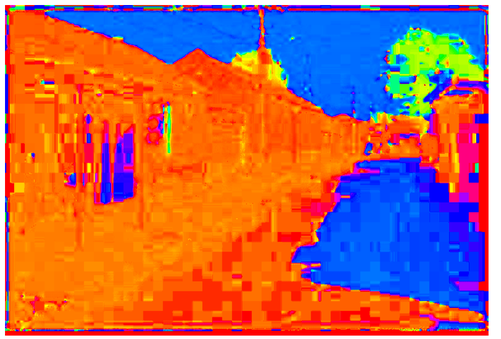


Espace de couleur : HSV

- *Hue* (teinte), *Saturation*, *Value* (intensité)

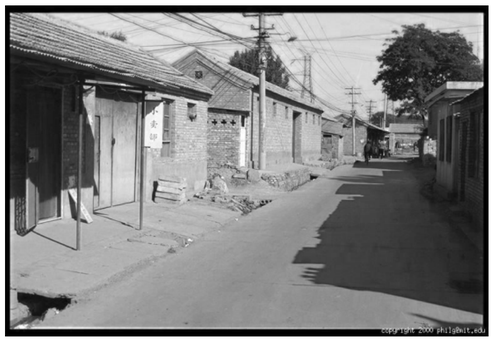
Très intuitif





H
(S=1,V=1)

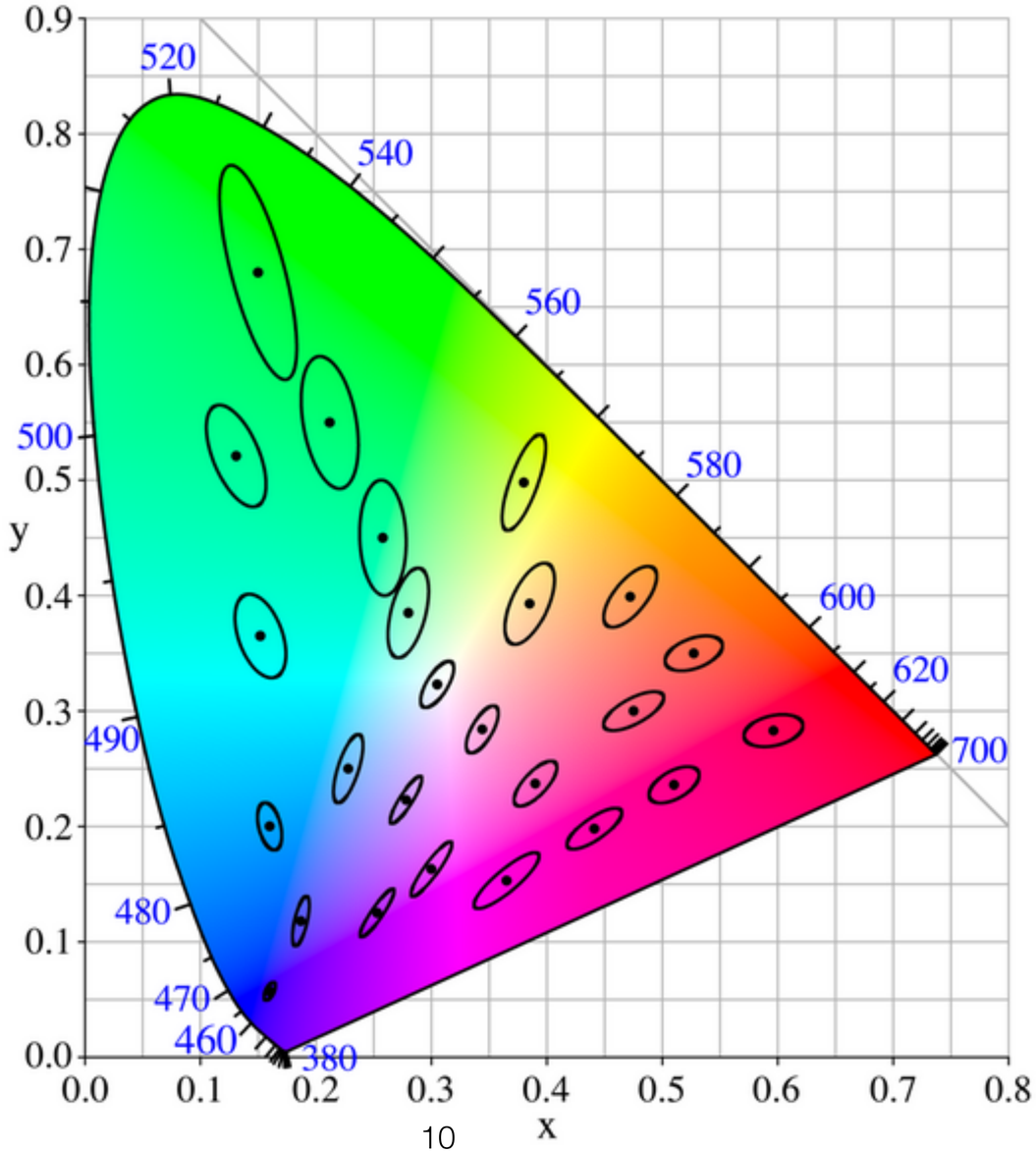
S
(H=1,V=1)



V
(H=1,S=0)

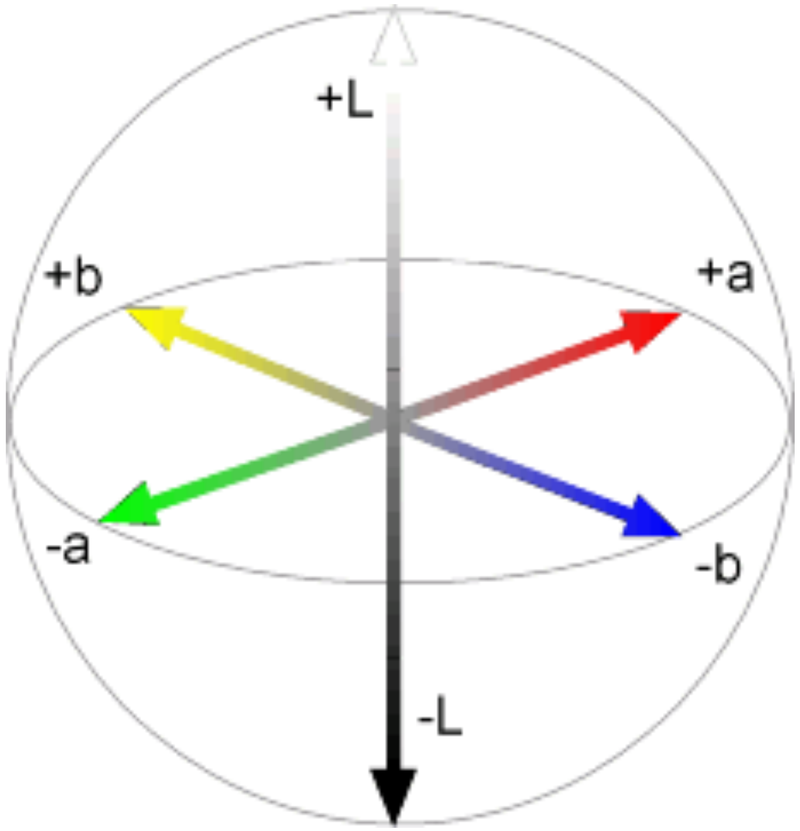
```
from skimage.color import rgb2hsv, hsv2rgb
```

« Uniformité perceptuelle »

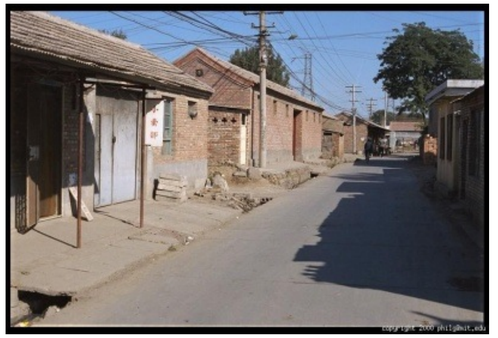


Espace de couleur : CIE L*a*b*

Espace de couleur “perceptuellement uniforme”



Luminance = intensité
Chrominance = couleur



L
(a=0,b=0)

a
(L=65,b=0)

b
(L=65,a=0)

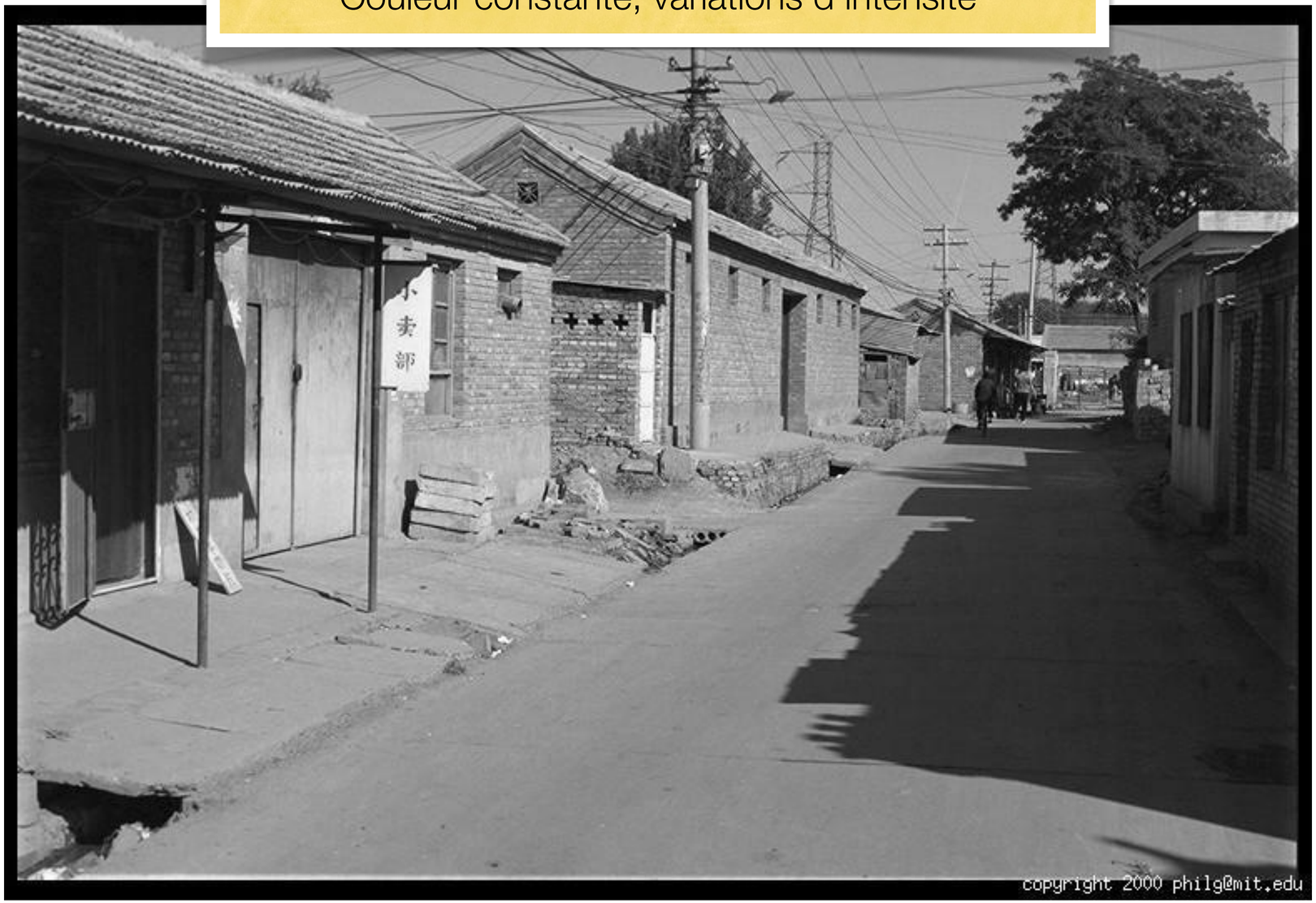
```
from skimage.color import rgb2lab, lab2rgb
```

Si vous pouviez choisir, garderiez-vous la luminance ou la chrominance?

Si vous pouviez choisir, garderiez-vous la
luminance ou la chrominance?

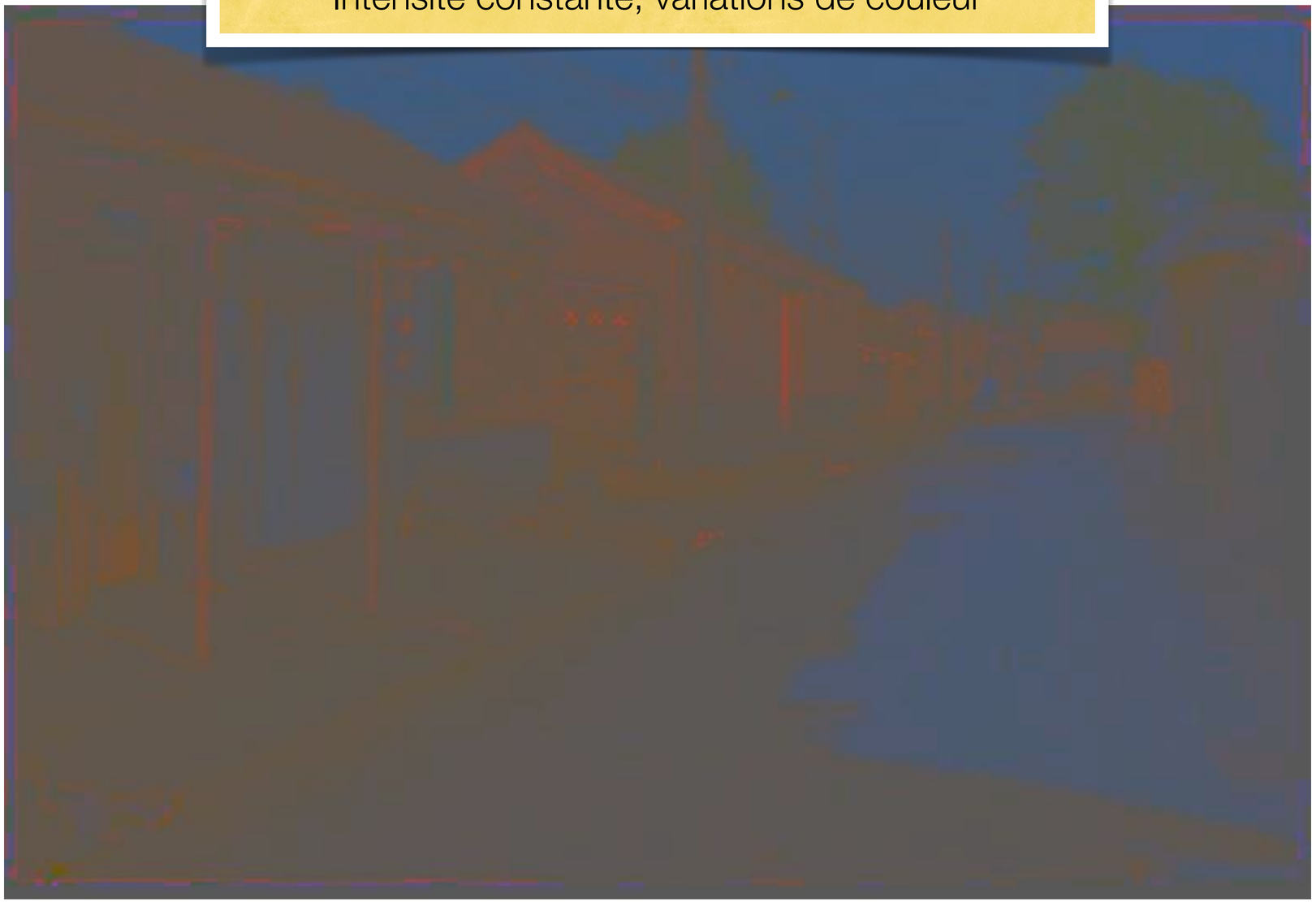
Plus d’information dans la **luminance**

Couleur constante, variations d’intensité

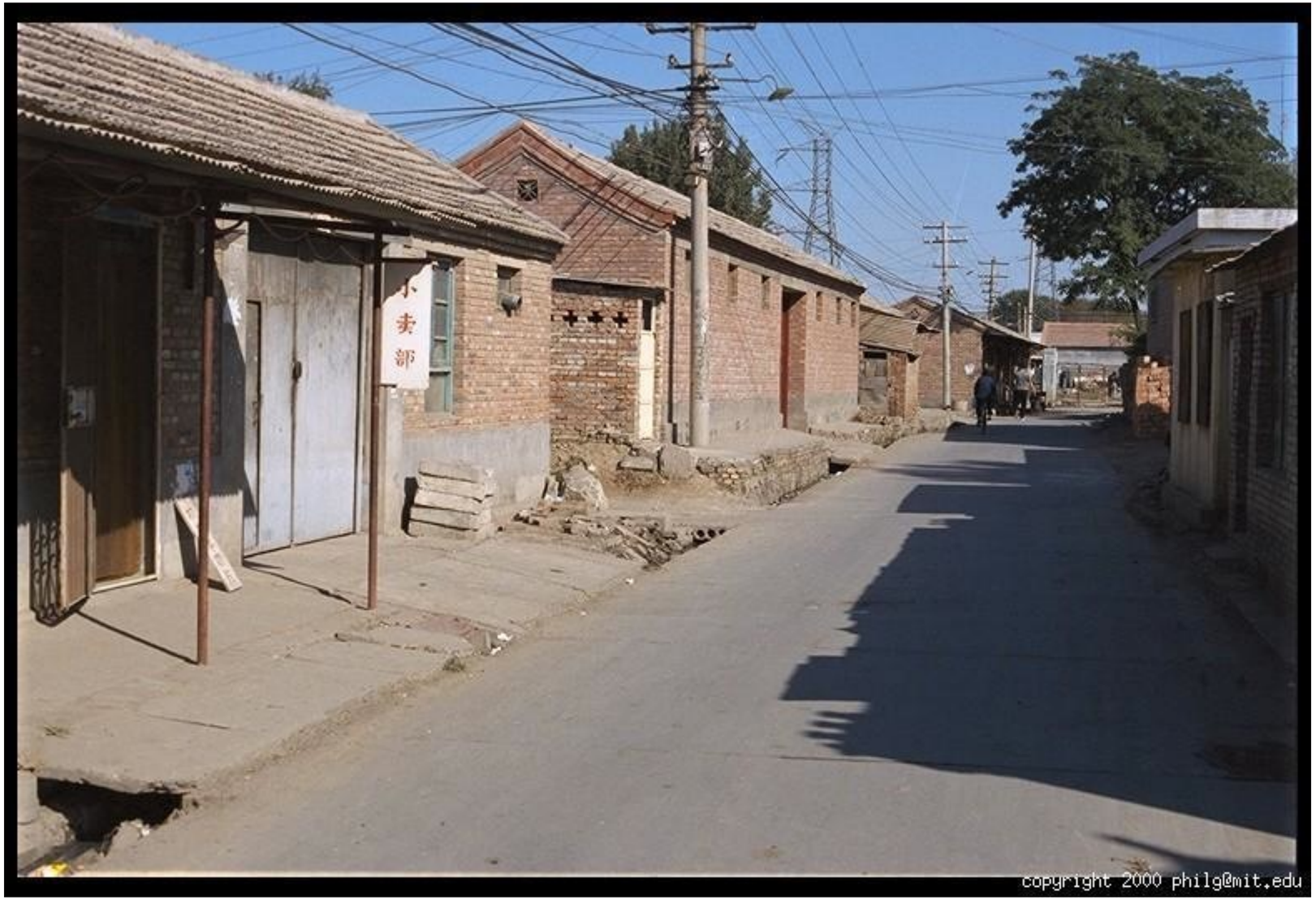


Plus d’information dans la **luminance**

Intensité constante, variations de couleur



Plus d’information dans la **luminance**



Compression (*chroma subsampling*)

Autre espace de couleur… Y’CbCr : Y’ (luma) et CbCr (chroma) mais l’idée reste la même

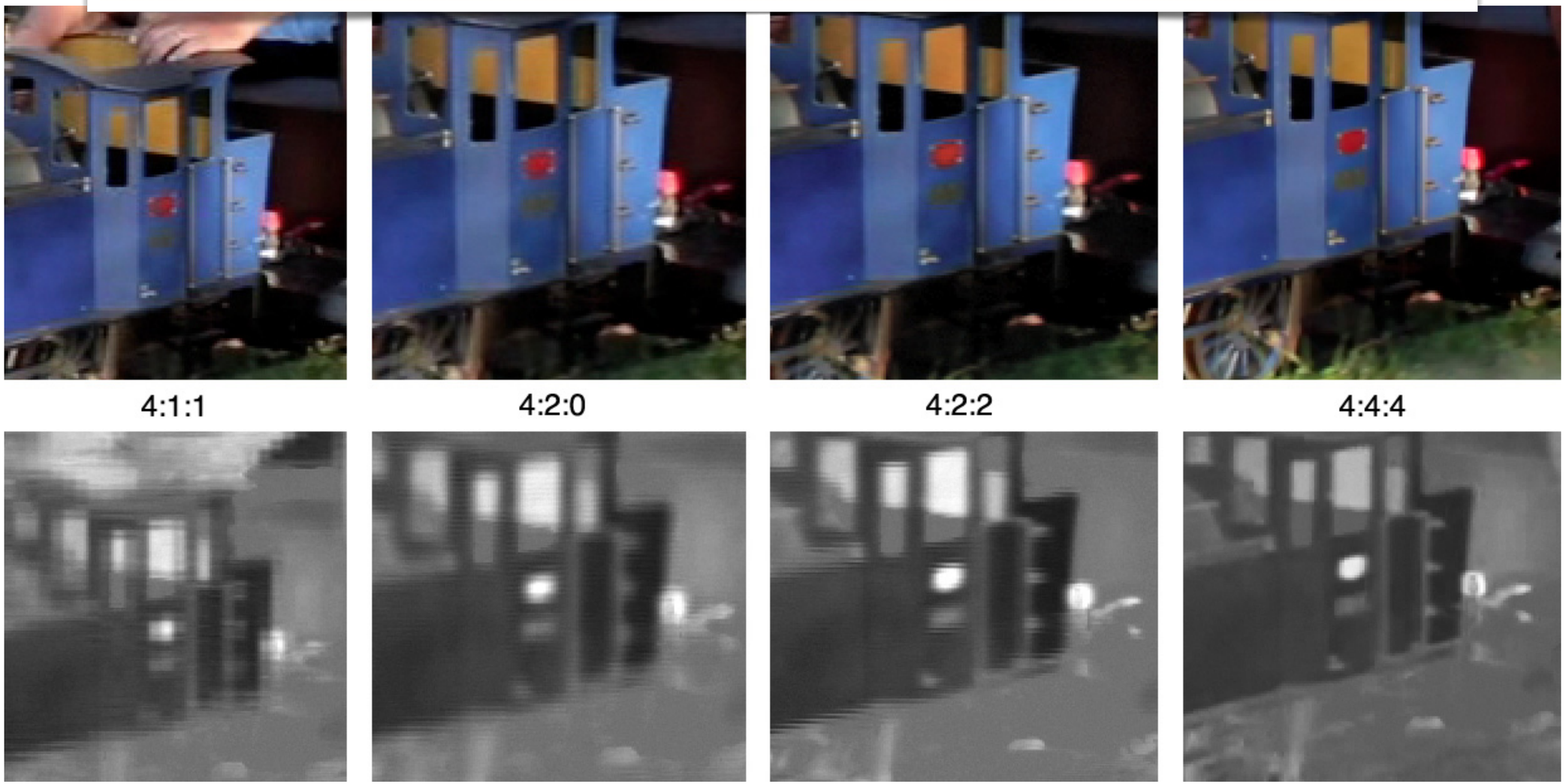
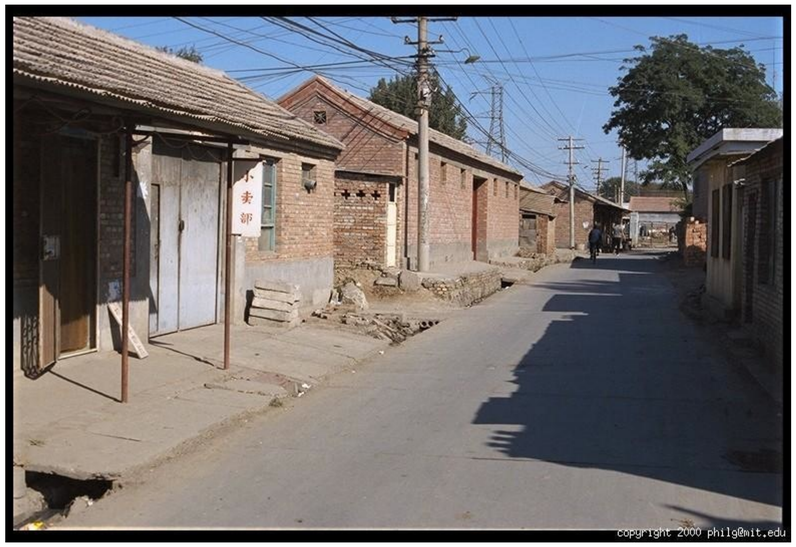


Image : source

Plus d’information dans la luminance

- Beaucoup d’algorithmes sont définis pour des images à un seul canal (intensité). Comment alors les appliquer à des images couleur?
  - Option #1 : appliquer à chaque canal de couleur indépendamment
  - Option #2 : convertir (ex. : en Lab), appliquer à la luminance en maintenant la couleur constante, re-convertir en RGB.

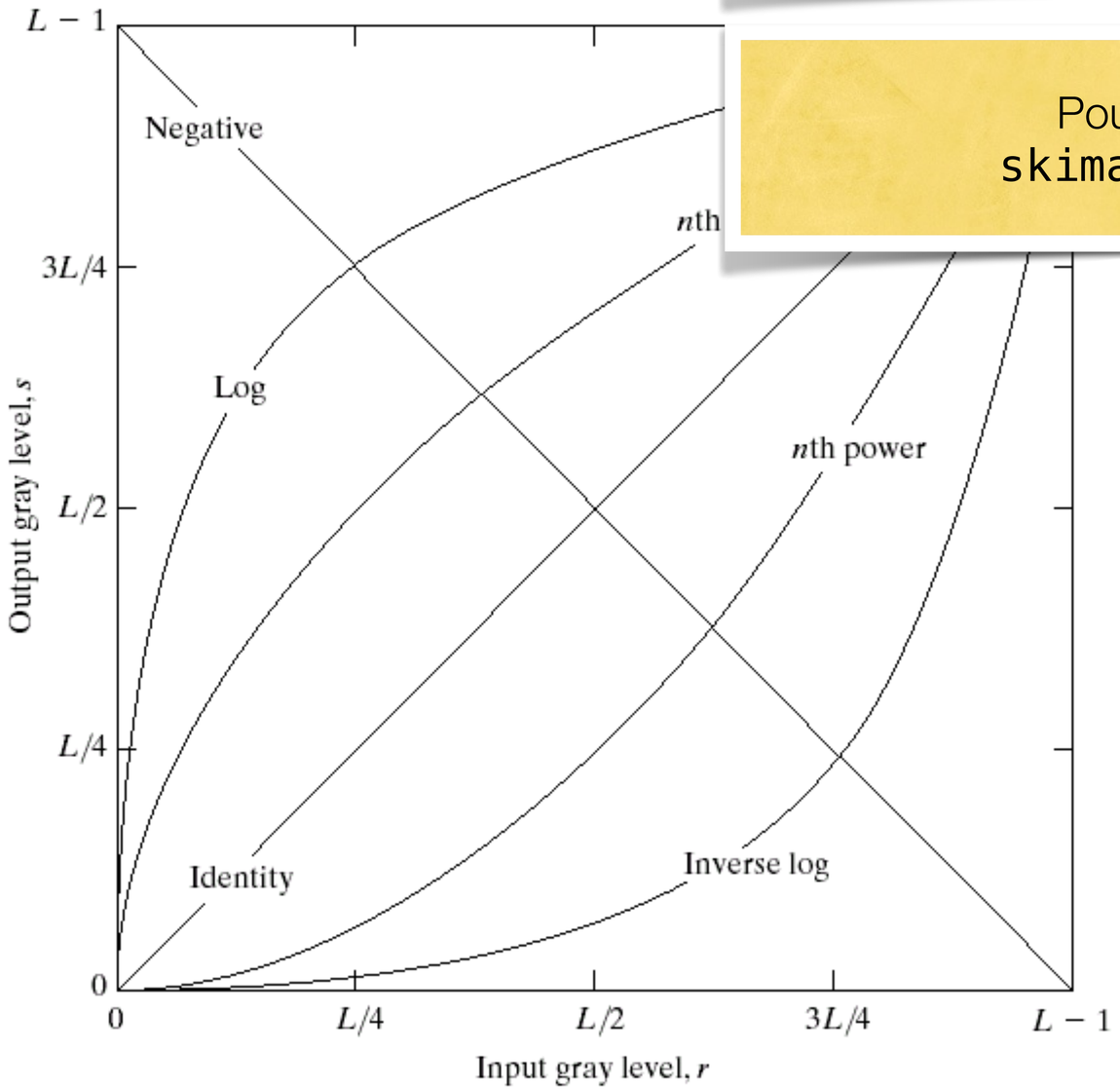
Pas de bonne ou mauvaise réponse absolue…
ça dépend des algorithmes!



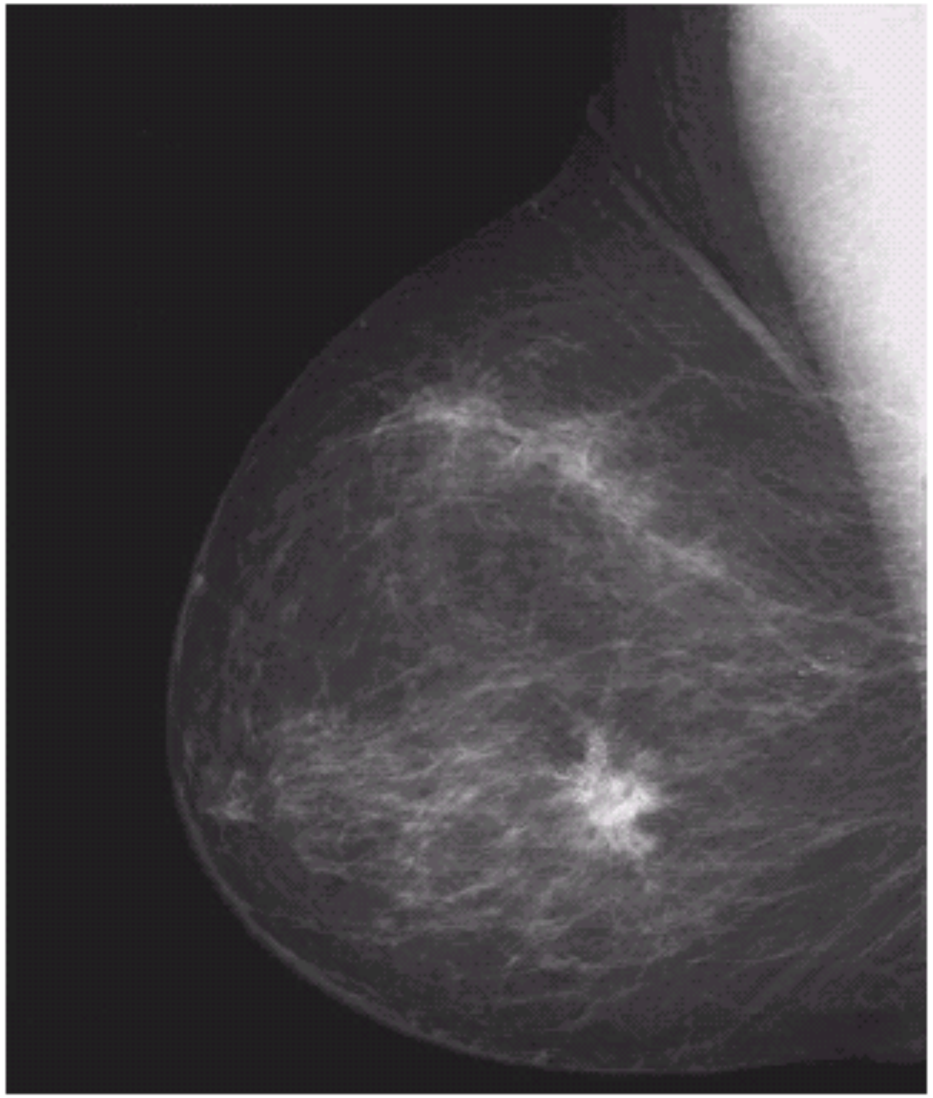
Transformée ponctuelles

Convertissez vos images en nombres à virgule (*float*)!
`skimage.img_as_float`

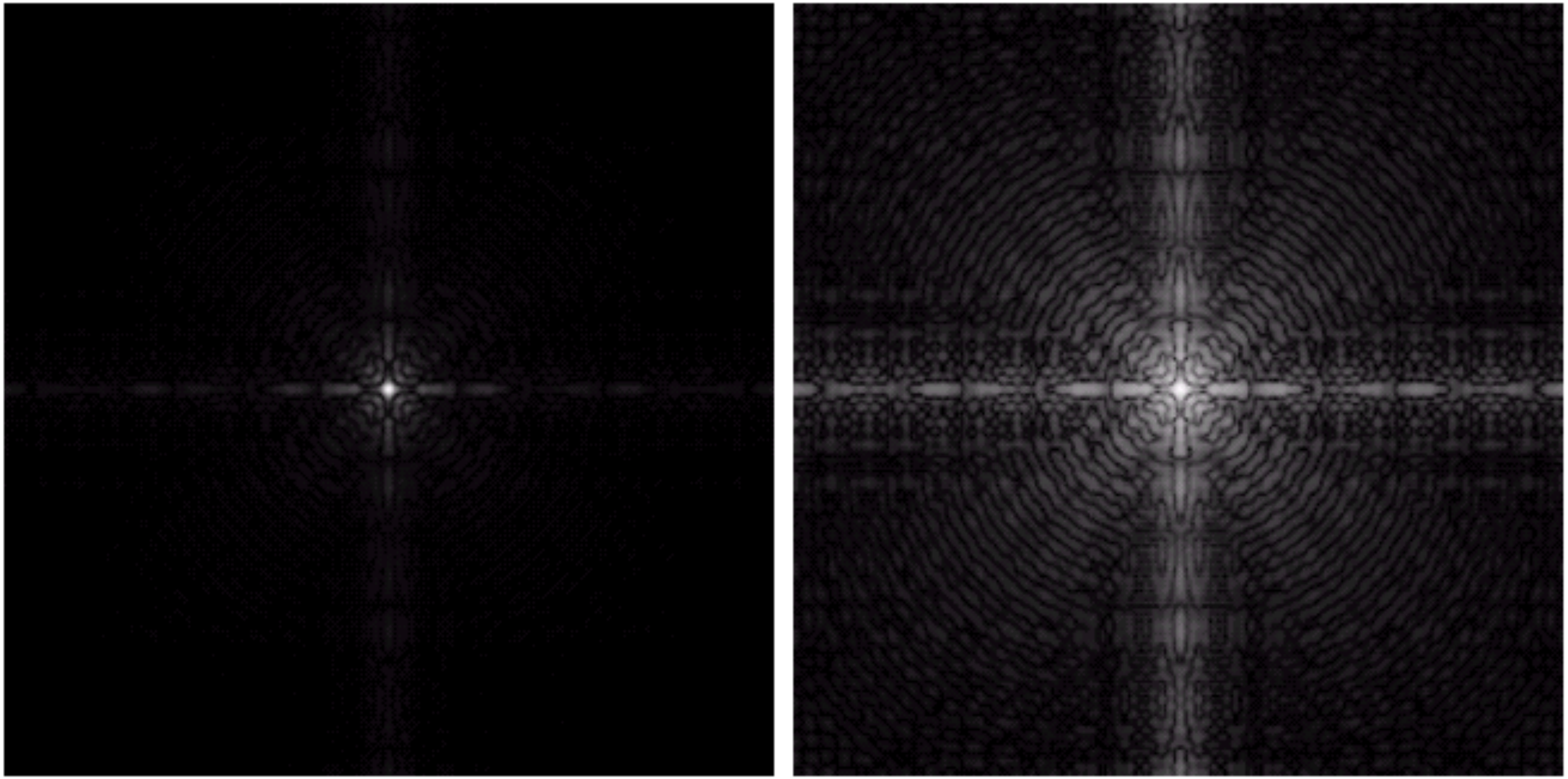
Pour revenir en uint8 :
`skimage.img_as_ubyte`



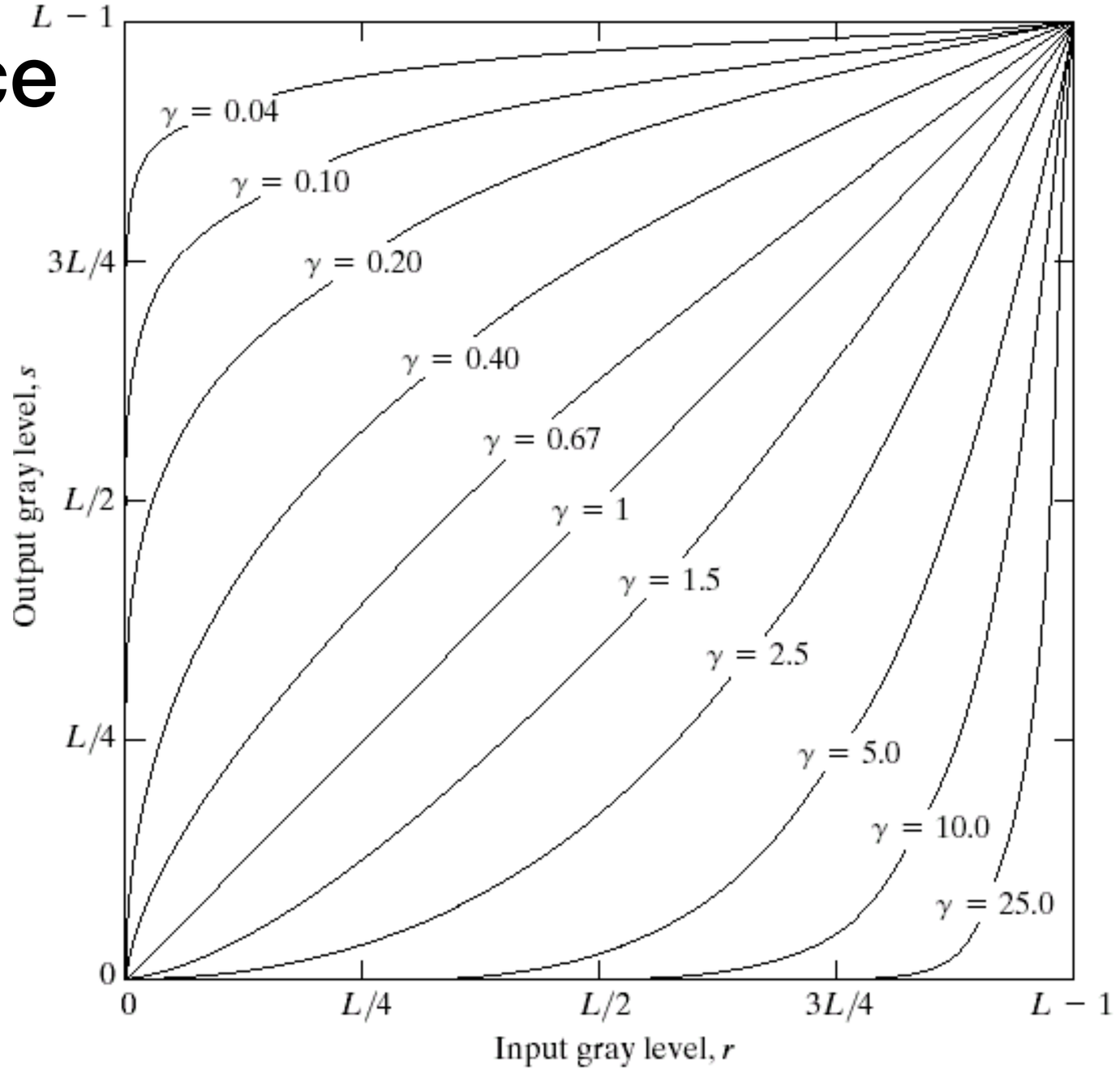
Négatif



Log



Puissance


L − 1
3L/4
L/2
L/4
0
Output gray level, s
γ = 0.04
γ = 0.10
γ = 0.20
γ = 0.40
γ = 0.67
γ = 1
γ = 1.5
γ = 2.5
γ = 5.0
γ = 10.0
γ = 25.0
0
L/4
L/2
3L/4
L − 1
Input gray level, r

“Clarification” d’images

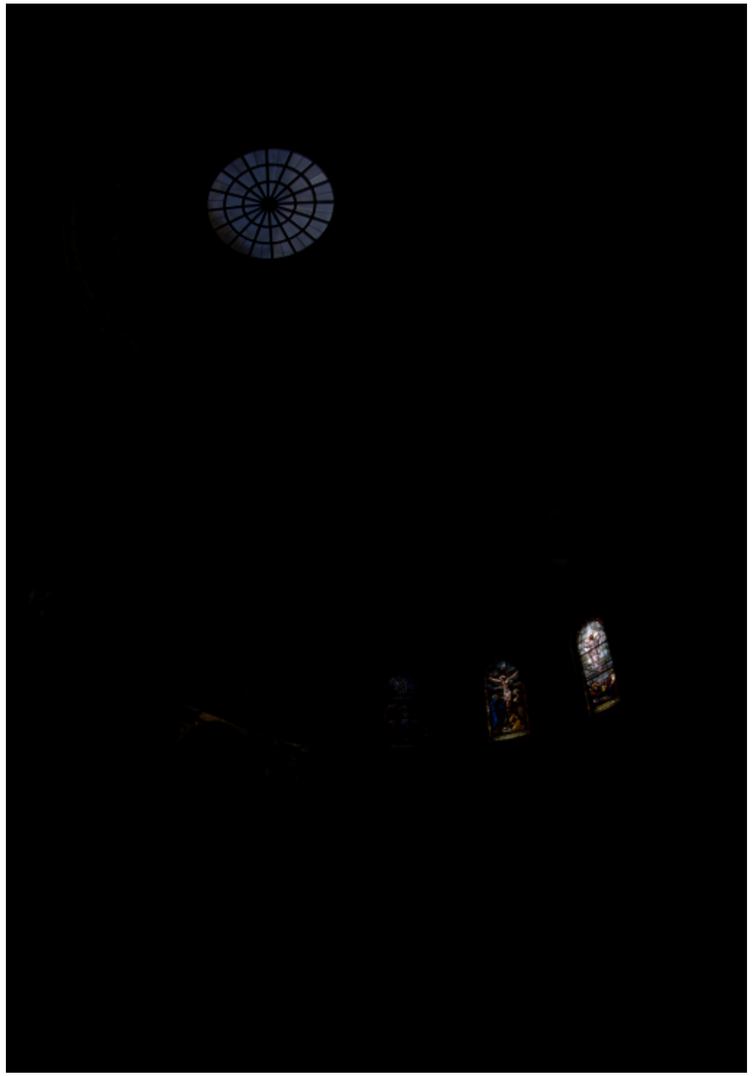
a b
c d

FIGURE 3.9
(a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)

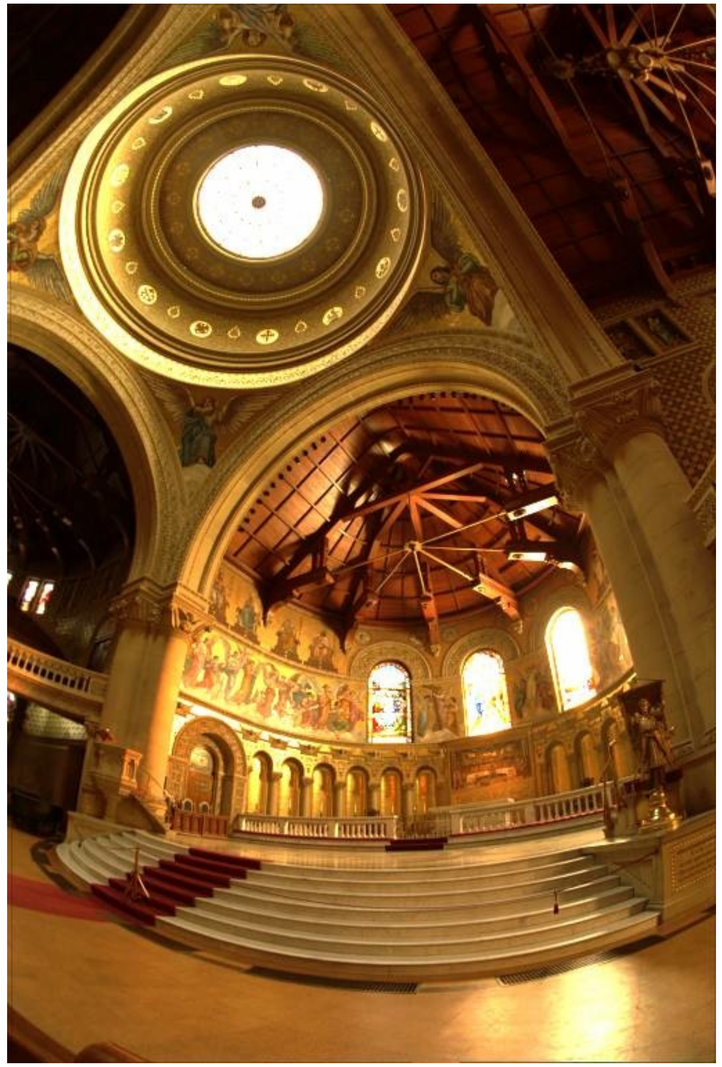
Reproduction tonale

- Problème typique :
  - Une caméra (de bonne qualité) peut capturer des valeurs d’intensité de 12 bits, mais nos écrans ne peuvent qu’en afficher 8 !

Linéaire



En fonction des pixels
les plus clairs

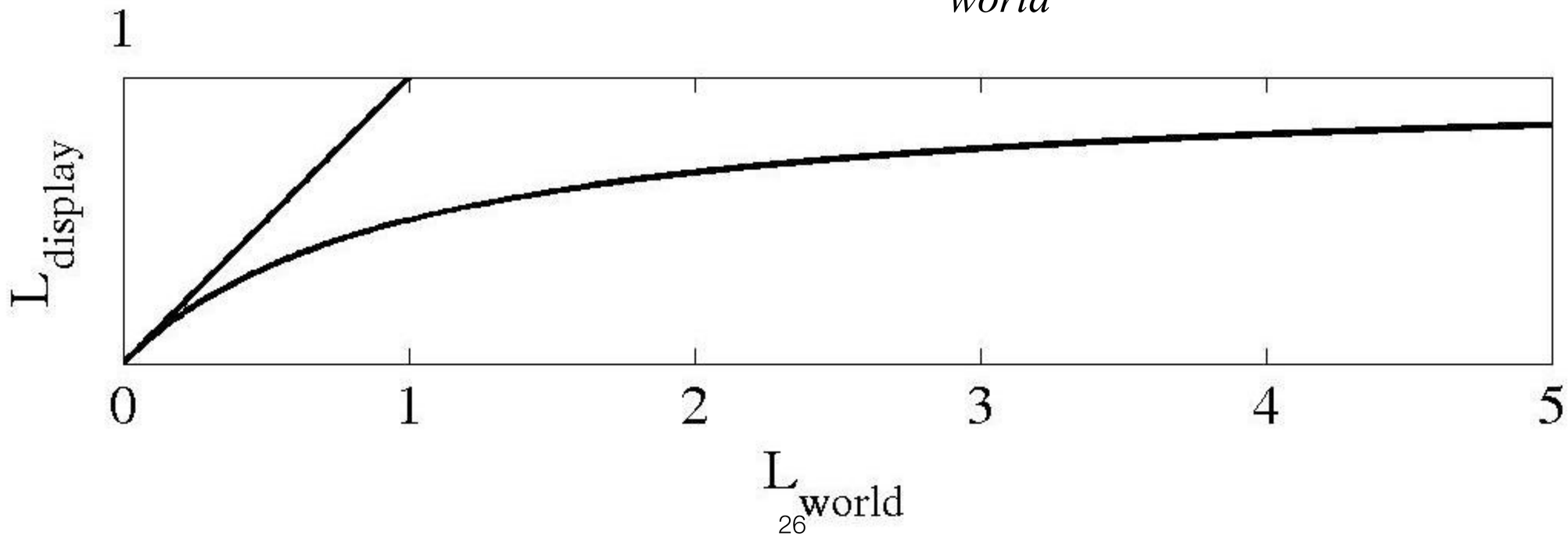


En fonction des pixels
les plus sombres

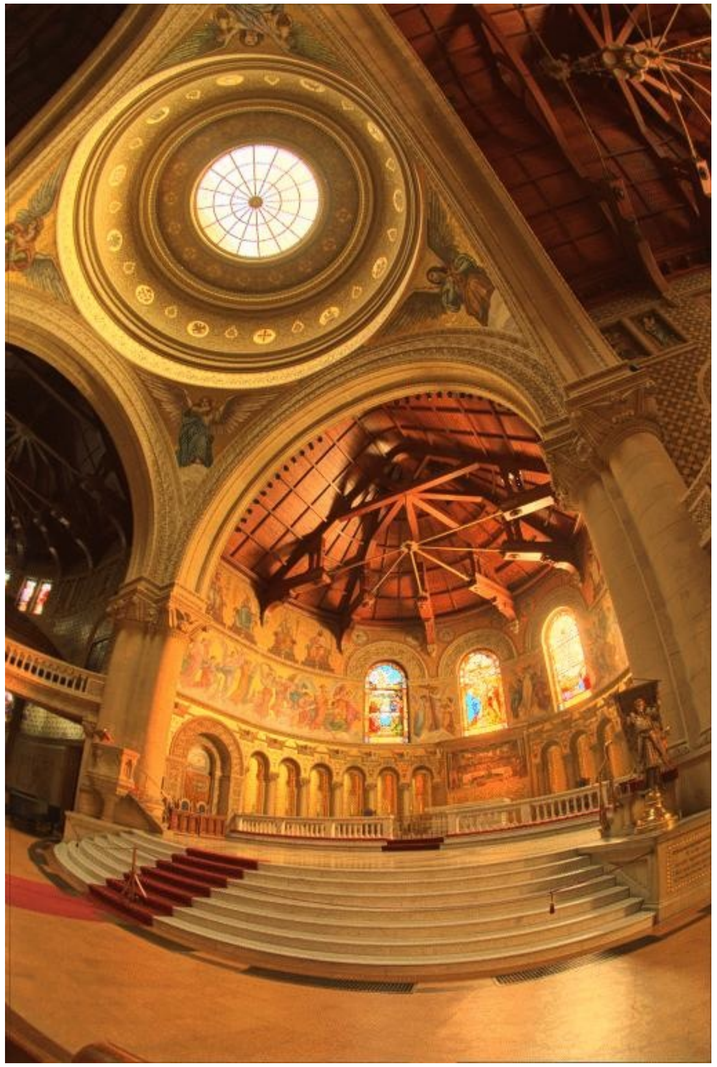
Opérateur global (Reinhard et al.)

- Solution simple : utiliser une transformée **non-linéaire**

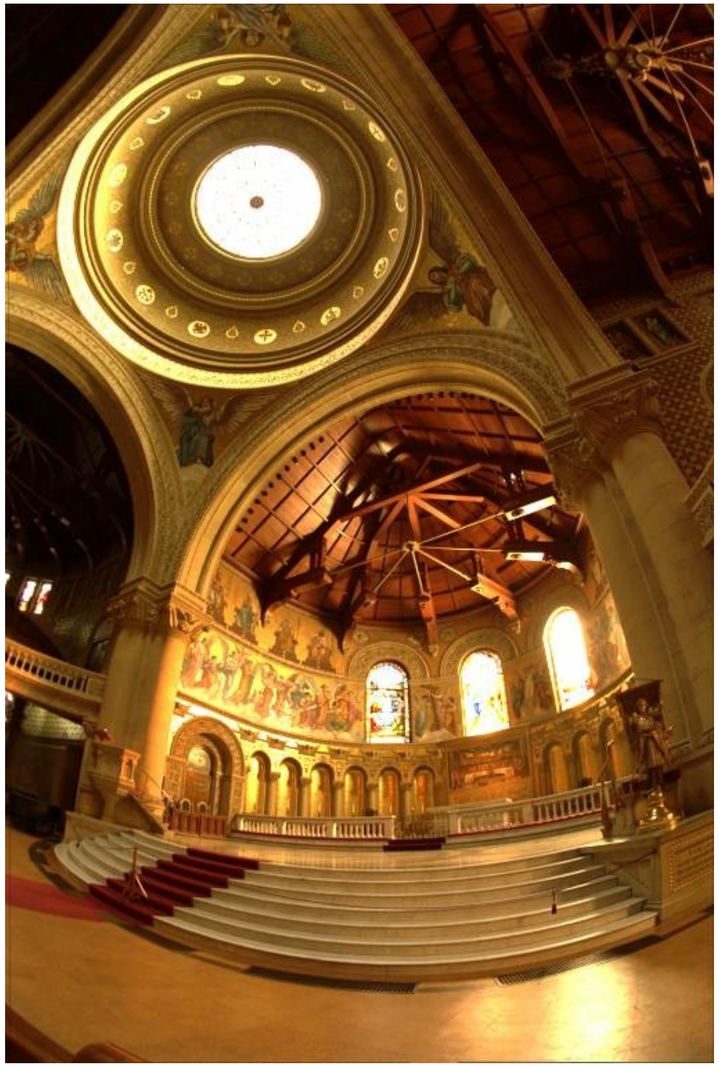
$$L_{display} = \frac{L_{world}}{1 + L_{world}}$$



Non-linéaire



Reinhard



En fonction des pixels
les plus sombres

Histogramme

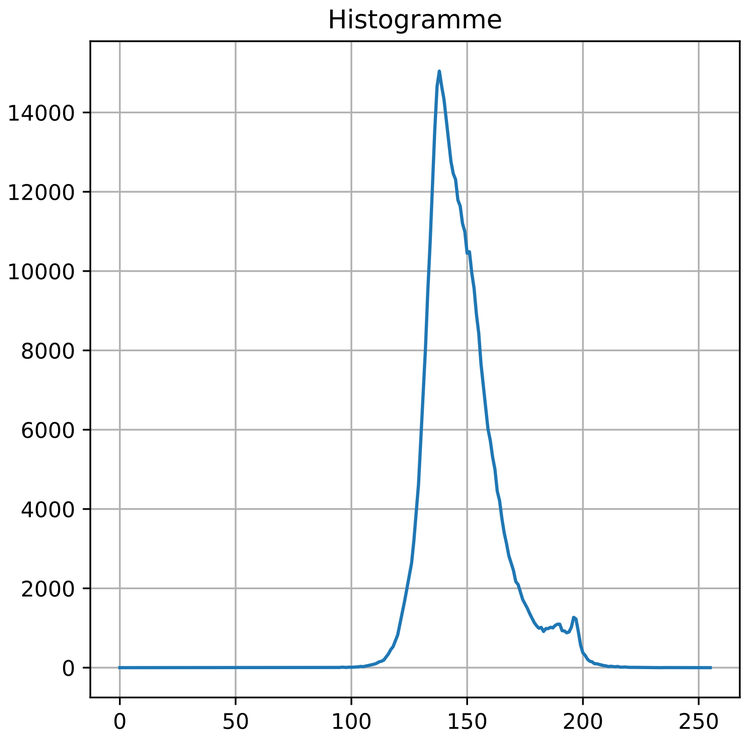
- Histogramme : compte le nombre de pixels qui ont chaque valeur d'intensité

$$h_i = \sum_{j \in pixels} \mathbf{1}(j == i)$$

- Histogramme cumulatif : combien de pixels ont au-moins une valeur d'intensité

$$c_i = c_{i-1} + h_i$$

Histogramme



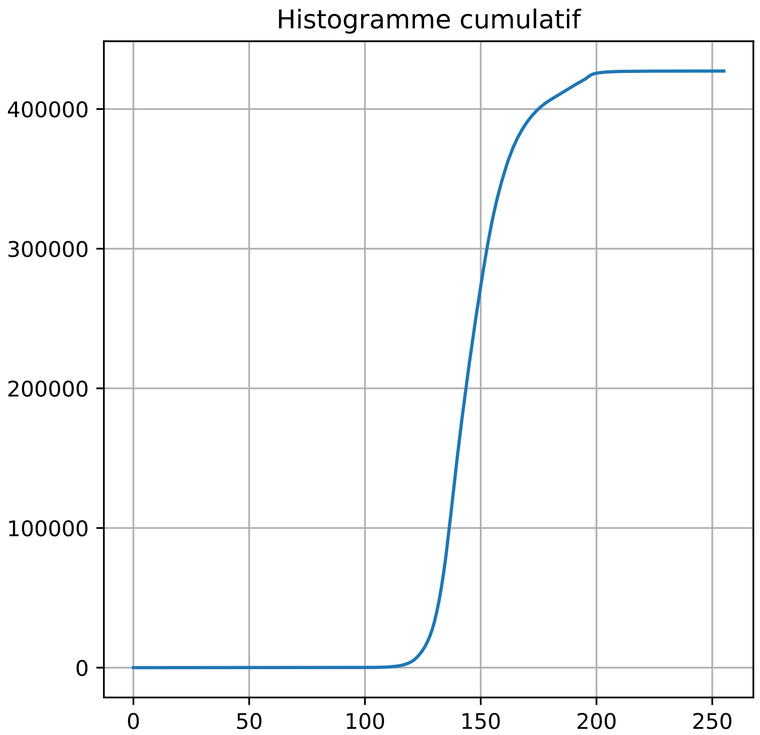
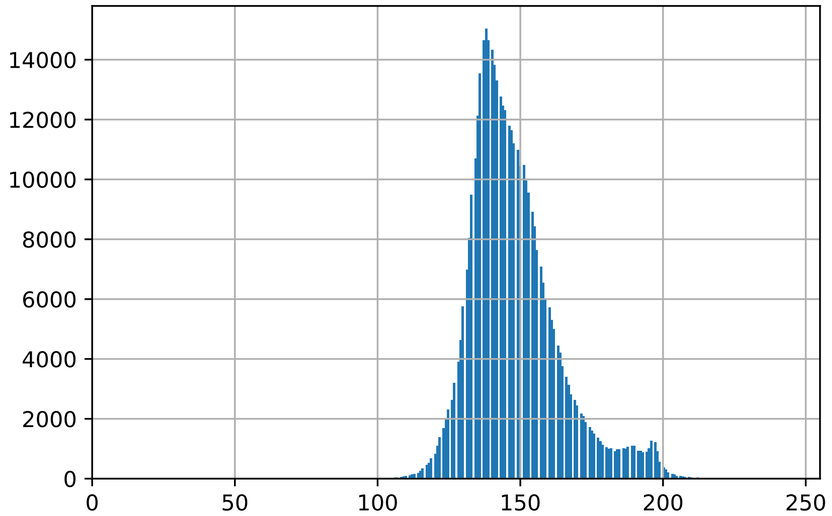


Image : source

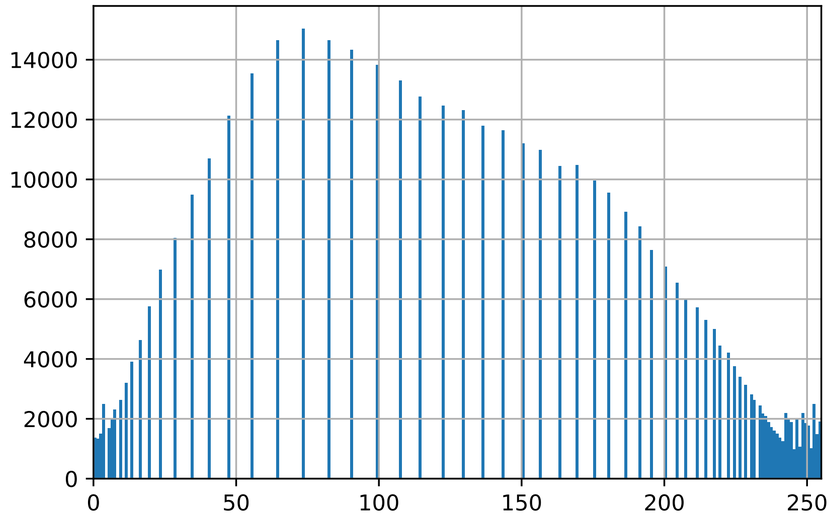
Égalisation d’histogramme

- Idée de base : ré-assigner les valeurs d’intensité de chaque pixel afin que l’histogramme de l’image résultante couvre la plage entre 0 et 255 le plus possible

Histogramme de l’image originale

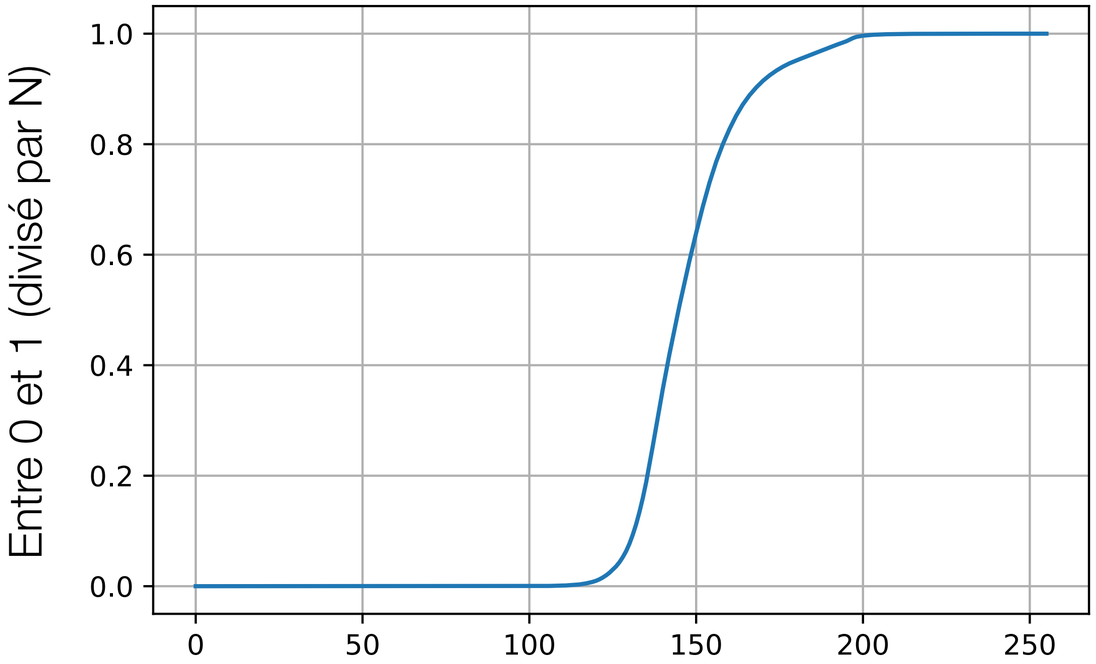


Histogramme égalisé



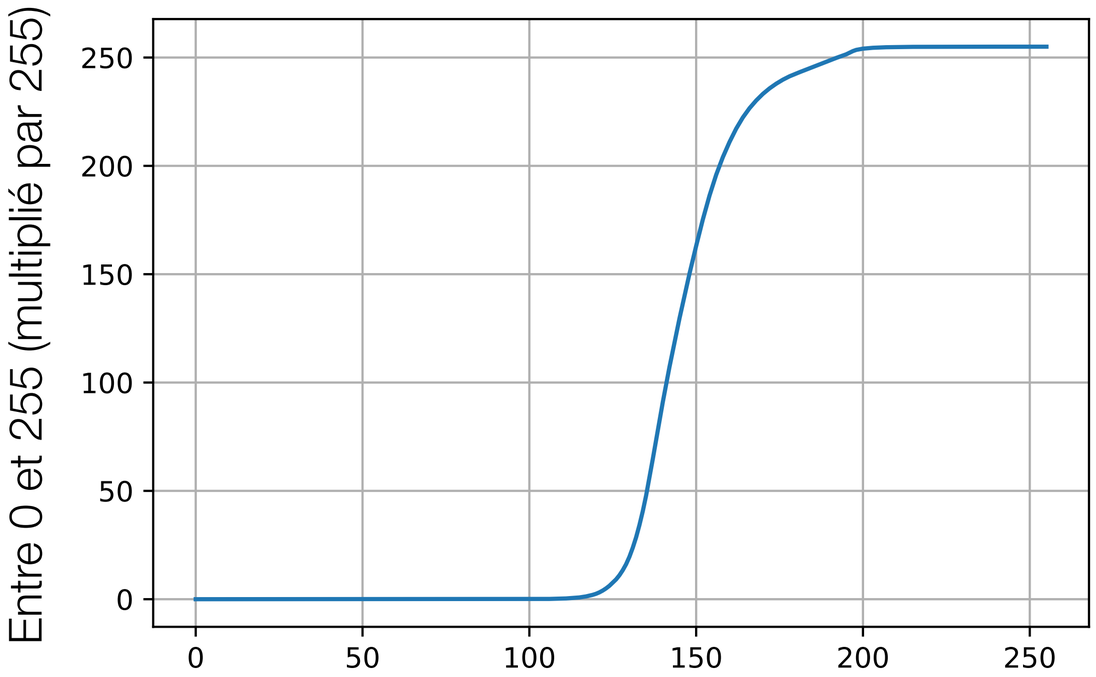
Égalisation d'histogramme : comment?

- Besoin d'une fonction $f(i)$ qui assigne une intensité i (entre 0 et 255) à une autre intensité
- Revenons à notre histogramme cumulatif…

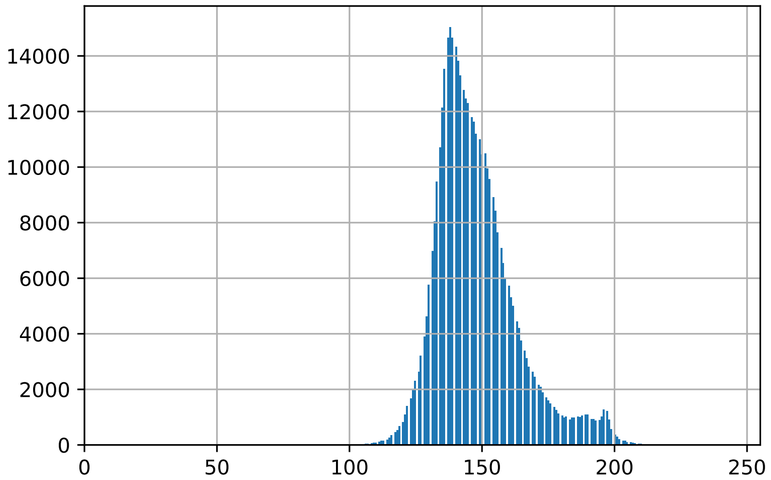


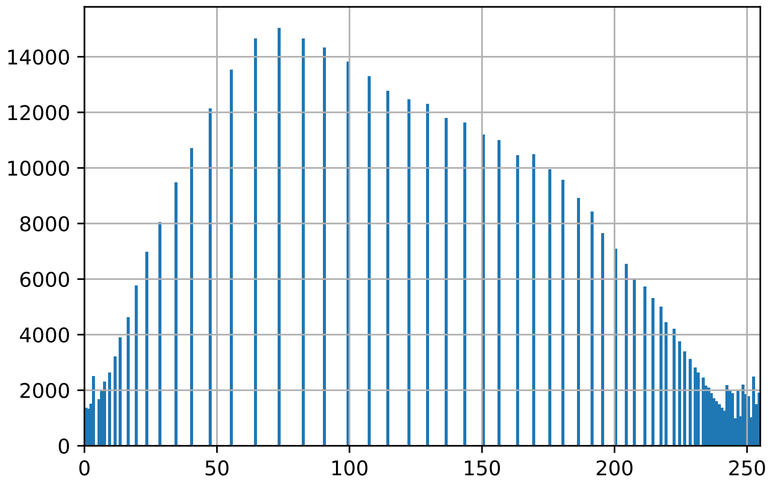
Égalisation d'histogramme : comment?

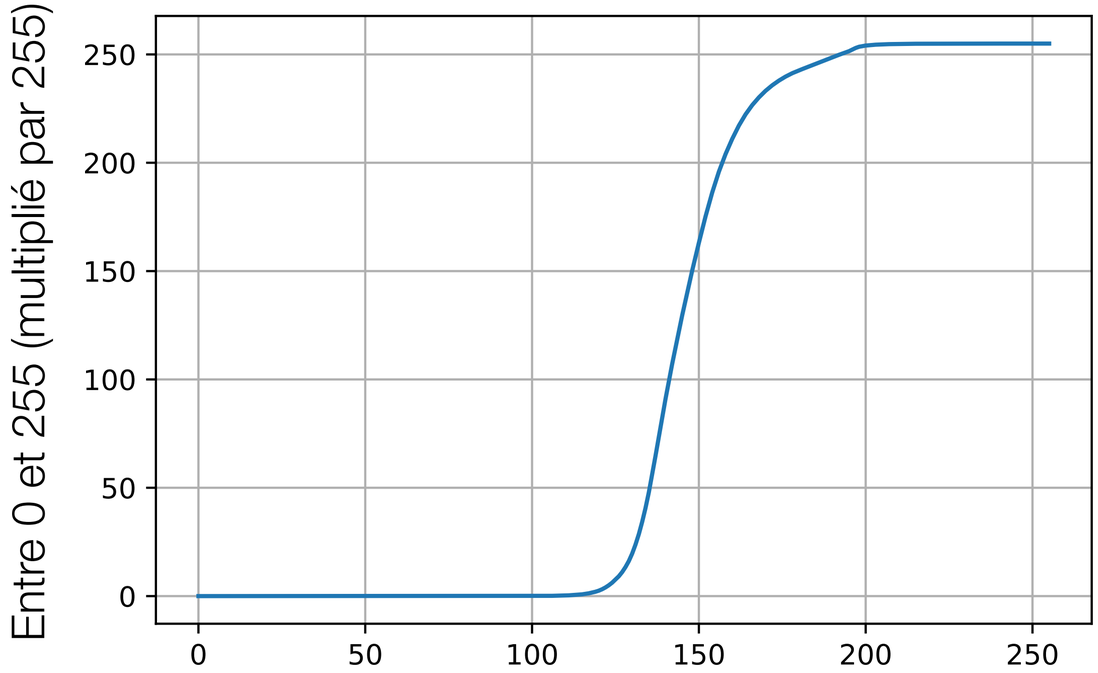
- Besoin d'une fonction $f(i)$ qui assigne une intensité i (entre 0 et 255) à une autre intensité
- Revenons à notre histogramme cumulatif…



Égalisation d’histogramme : comment?







Égalisation d'histogramme : algorithme

- Besoin d'une fonction $f(i)$ qui assigne une intensité i (entre 0 et 255) à une autre intensité

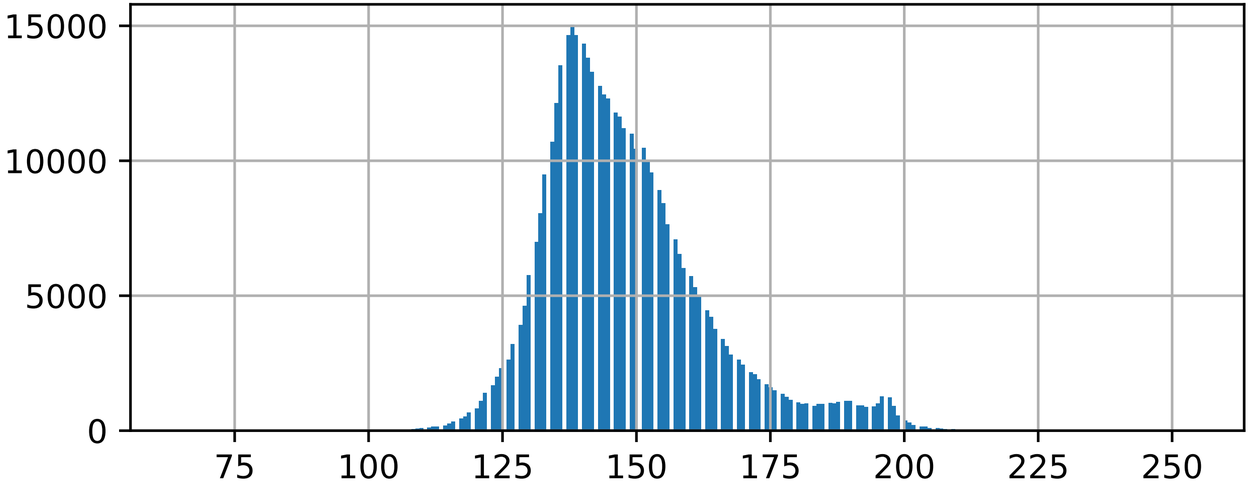
1. Calculer l'histogramme cumulatif $c(i)$
2. $f(i) = \dfrac{c(i) \times 255}{N}$
3. Combiner avec l'image originale (pour contrôler le niveau d'égalisation)

$$f(i) = \alpha \frac{c(i)}{N} 255 + (1 - \alpha) i$$

Contraste

Image

Histogramme (image originale)



imgEq

Histogramme (image égalisée)

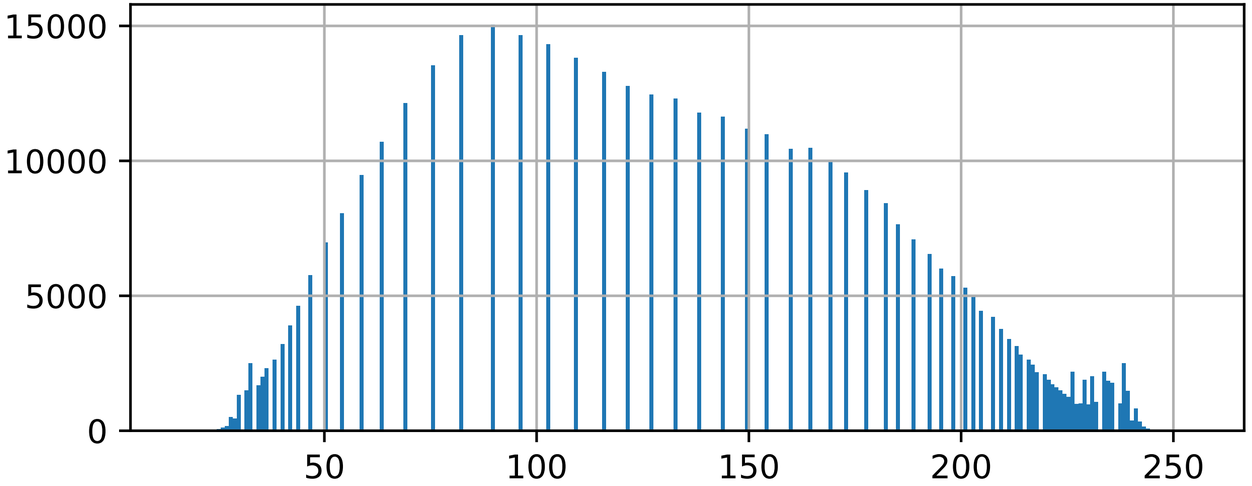


Image : source

Question

Comment égaliser l’histogramme d’une image couleur?

Essayez-le!

La métaphore du photomètr

Vrai ou faux?

La façon dont nous percevons la couleur est déterminée
entièrement par le spectre de lumière qui illumine chaque récepteur
(comme un photomètre)

Qu'en est-il d'une caméra?

FAUX : nos yeux **s'adaptent** aux conditions ambiantes!
C'est ce que l'on appelle la **constance de couleur**.

© Stephen E. Palmer, 2002

Équilibrage des blancs (*white balance*)

Image originale

Image ajustée

Image : source

Dans une caméra

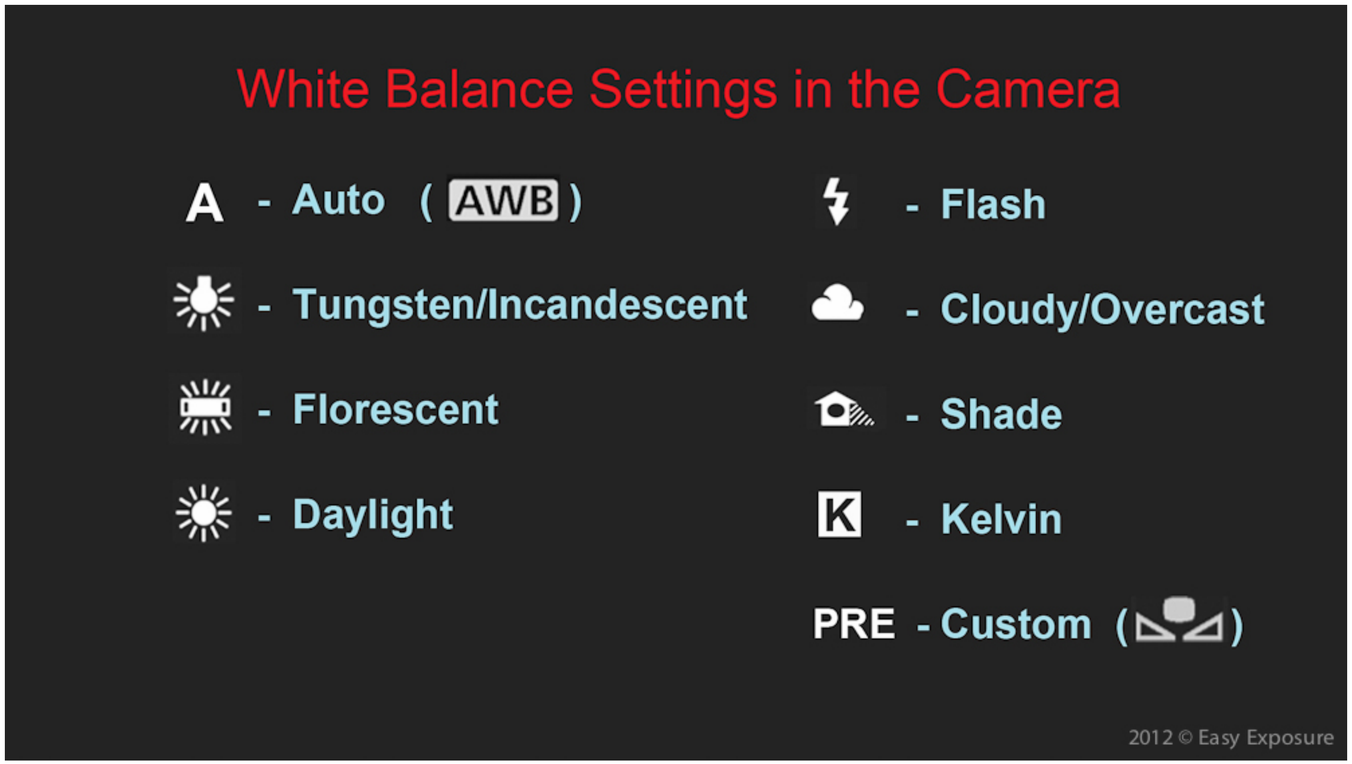


Image : source

Ajustement après avoir pris la photo

- Manuellement : sélection d’un objet neutre dans la scène

Algorithme

- Demander à l'utilisateur de cliquer sur un objet qui devrait être neutre, obtenir sa `couleur`
- Calculer un `décalage` à appliquer à chaque canal pour que `couleur` soit neutre (R = G = B)
- Appliquer ce `décalage` à tous les pixels de l'image

Ajustement automatique?

- *Automatic white balance*, ou *AWB*
  - *Grey world* : moyenne de la scène est grise
  - *White world* : objet le plus brillant est blanc