

---

CES PAGES SONT DESTINÉES À FACILITER LA PRISE DE NOTES. ELLES SONT PRODUITES AUTOMATIQUEMENT À PARTIR DU SOURCE L<sup>A</sup>T<sub>E</sub>X/BEAMER DU DIAPORAMA SERVANT DE SUPPORT DE COURS. LA PRÉSENTATION ET LA MISE EN PAGE SONT PAR CONSÉQUENT APPROXIMATIVES.

---

## Partie C — thème 1

### Quantification d'image



256 niveaux



6 niveaux



162263 couleurs



8 couleurs

## 1 Introduction

### 1.1 Objectifs de la quantification

- discrétiser une information *continue* de luminance ou de couleurs :

$$\begin{aligned} [0, 1] &\longrightarrow [0 .. M - 1] \\ \text{ou } [0, 1]^3 &\longrightarrow [0 .. M - 1]^3 \end{aligned}$$

- ramener la *dynamique* d'une image (discrète) dans la dynamique de l'affichage
  - *visualisation de données scientifiques* (ex. : spectre de fourier)

$$[0 .. X_{max}] \longrightarrow [0 .. 255]$$

- *format raw*  $\longrightarrow$  *format écran* (ex. : photo couleur numérique)

$$[0 .. 2^{16} - 1]^3 \longrightarrow [0 .. 255]^3$$

- réduire le nombre de niveaux / couleurs d'une image affichable

$$[0 .. 255] \longrightarrow [0 .. M - 1], \text{ avec } M \text{ petit}$$

- réduire la *taille* de l'image
- *agréger les pixels voisins visuellement proches* (effets spéciaux, segmentation)

## 1.2 Principe de mise en œuvre

— images monochromes

— partitionner l'ensemble des niveaux de gris en  $M$  classes d'équivalences

exemple :

$i$	[0..10]	[11..68]	[69..83]	[83..142]	[143..255]
$c_i$	0	1	2	3	4

— choisir un représentant  $\bar{c}_i$  pour chaque classe  $c_i$

exemple :

$i$	[0..10]	[11..68]	[69..83]	[83..142]	[143..255]
$\bar{c}_i$	5	12	75	138	199

— fonction de quantification  $Q(i)$  : pour chaque pixel  $p$ ,  $I'(p) = \overline{c_{I(p)}}$

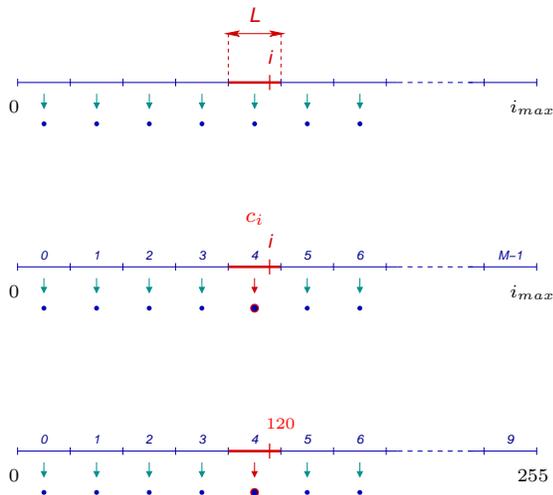
— images couleur : même principe mais nécessite généralement un traitement vectoriel

— quantification uniforme (prédéfinie) ou adaptative (dépendant de l'image)

## 2 Quantification Uniforme

### Images monochrome

— découpage de  $[0..i_{max}]$  en  $M$  classes de même taille  $L$



$$L = \frac{i_{max} + 1}{M}$$

— classe de  $i$  (numérotée à partir de zéro)

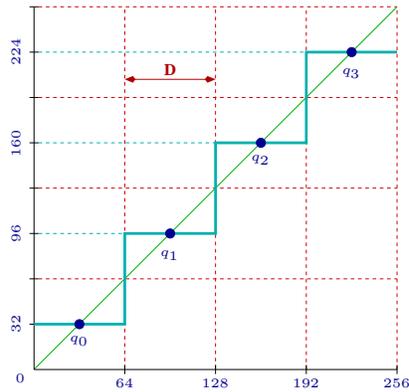
$$c_i = \left\lfloor \frac{i}{L} \right\rfloor$$

exemple  $i_{max} = 255$  et  $M = 10$ ; largeur :  $L = 25.5$

$$120 \rightarrow \frac{120}{25.5} = 4.7059 \rightarrow 4$$

— choisir le représentant de la classe  $c_i$

$$Q(i) = \left[ c_i \times L \right] + \frac{L}{2}$$



[0..63]	→	32
[64..127]	→	96
[128..191]	→	160
[192..255]	→	224

$$Q(i) = \left[ \left[ \frac{i}{L} \right] \times L + \frac{L}{2} \right], \text{ avec } L = \frac{i_{max} + 1}{M}$$

— Exemple :



256 niveaux



8 niveaux



4 niveaux

### Images couleur

— quantification uniforme : marginale (effectuée indépendamment sur chaque canal)



256 niveaux



4 × 4 × 4 couleurs (39 utilisées)



2 × 2 × 2 couleurs

## 3 Loi de Weber-Fechner

La quantification uniforme n'est pas adaptée à la perception humaine



E. H. Weber

$$\frac{\Delta I}{I} = \text{cte}$$

- loi formulée par Gustav Fechner (1801-1887) à partir des travaux expérimentaux d'Ernst Weber (1795-1878)
- étude quantitative du lien entre stimulus physiques et sensation (comparaisons de poids, de sons, de teintes)
- plus petite différence perceptible = fraction constante de l'excitant



### Organiser les échantillons en progression géométrique

- correction logarithmique avant quantification uniforme



$$y = \lambda_1 \log_b(1 + \lambda_2 x) + \lambda_3$$

ex. : transformation  $[0, 1] \rightarrow [0, 1]$  (adaptée aux images sombres) :  $\frac{1}{8} \log_2(1 + 255x)$

- correction en loi de puissance :  $x^{\frac{1}{\alpha}}$

— par exemple pour les images naturelles :

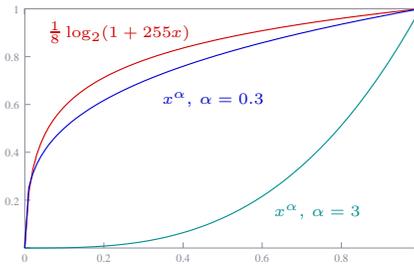
$\simeq \frac{1}{3}$  (env. sombre)

↑

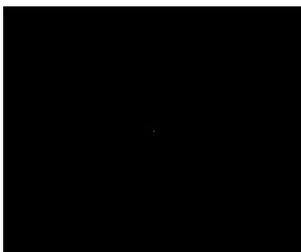
$\frac{1}{\alpha}$

↓

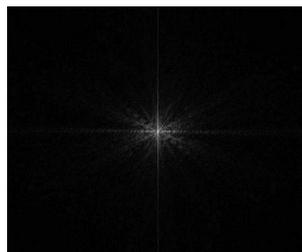
$\simeq 3$  (env. très clair)



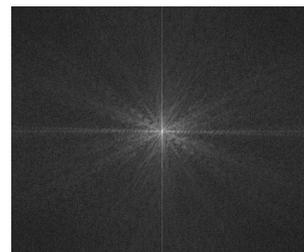
### Exemple : visualisation de spectre d'amplitude



quantification uniforme  
(dynamique  $> 10^6$ )



idem après correction logarithmique

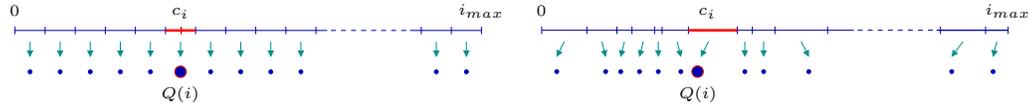


idem après correction  $x^{0.2}$

## 4 Quantification adaptative

### 4.1 Quantification optimale

- recherche d'une fonction  $Q_I(i)$  adaptée à chaque image  $I$  : optimiser la **définition des classes** et le **choix des représentants**



- objectif : minimiser l'écart entre  $I$  et  $Q_I(I)$  : **différence quadratique**

● = (236, 204, 95)

$I_1$

● = (205, 148, 97)

$I_2$

$$\begin{aligned}
 & (236 - 205)^2 \\
 & + (204 - 148)^2 \\
 & + (95 - 97)^2 \\
 & = 4101
 \end{aligned}$$

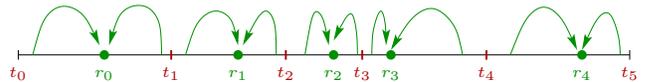
$$DQ(I_1, I_2) = \sum_{p \in D} (I_1(p) - I_2(p))^2$$

### 4.2 Images monochromes : quantification sous-optimale

- **découpage** de l'ensemble des  $N$  niveaux de gris en  $M$  intervalles  $[t_i, t_{i+1}]$

$$\begin{aligned}
 & [t_0, t_1] \\
 & [t_1, t_2] \\
 & \vdots \\
 & [t_{M-1}, t_M]
 \end{aligned}$$

avec  $t_0 = 0 < t_1 < \dots < t_M = N-1$



- choix d'un **représentant**  $r_j$  pour chaque intervalle

$$\forall i \text{ t.q. } t_j \leq i < t_{j+1}, Q(i) = r_j$$

- trouver les  $\{t_j\}$  et  $\{r_j\}$  **minimisant la différence quadratique** entre  $I$  et  $Q(I)$

- on peut montrer que les  $\{t_j\}$  et  $\{r_j\}$  doivent satisfaire deux conditions :

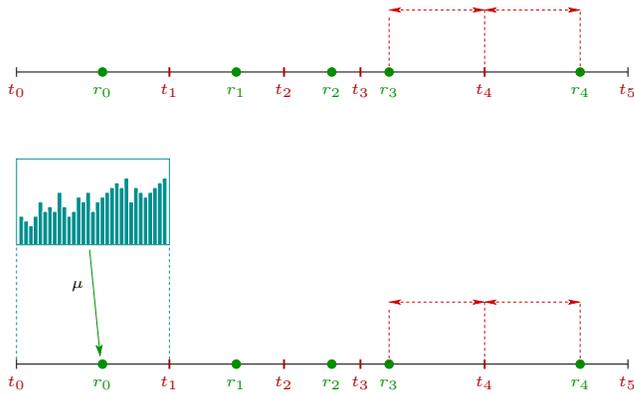
- $t_i$  est le **milieu** du segment  $\overline{r_{i-1}r_i}$

$$t_i = \frac{r_{i-1} + r_i}{2}, \quad i = 1, 2, \dots, K-1$$

- $r_i$  est la **moyenne** de tous les pixels  $p$  tels que  $I(p) \in [t_i, t_{i+1}]$

$$r_i = \sum_{j=t_i}^{t_{i+1}-1} j h'(j), \quad i = 1, 2, \dots, K$$

où  $h'$  est l'histogramme normalisé de l'image



— conditions nécessaires mais non suffisantes : sous-optimal

### Exemple avec l'image des éoliennes pour $M = 5$



— initialisation des  $t_i$  de manière équirépartie

— algorithme de Lloyd-Max : répéter

1. calcul des  $r_i$  en fonction des  $t_i$  (moyennes)
2. calcul des  $t_i$  en fonction des  $r_i$  (milieux)

jusqu'à la stabilité des données

— solution sous-optimale mais convergence rapide

### Exemples



256 niveaux



5 niveaux

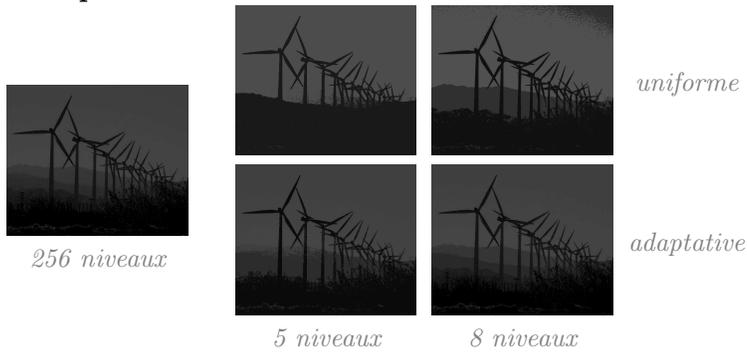


8 niveaux

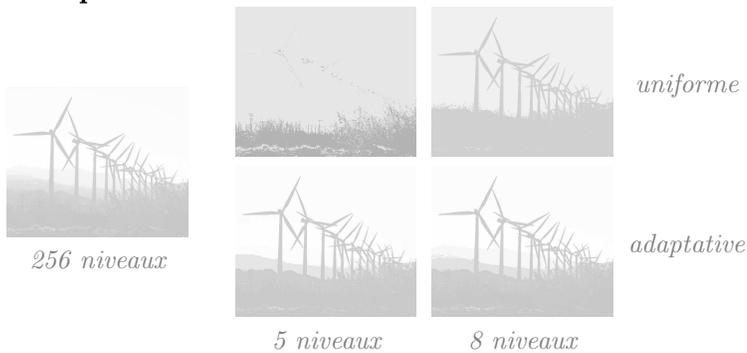
uniforme

adaptative

## Exemples



## Exemples



## 4.3 Quantification adaptative d'images couleur

### Quantification optimale d'image couleur

- même principe qu'en dimension 1 :
- partitionner l'ensemble  $C$  des couleurs de  $I$  en  $n$  classes d'équivalence

$$C = C_1 \cup \dots \cup C_n \quad \text{et} \quad C_i \cap C_j = \emptyset$$

- choisir les *représentants* des classes :  $\{\bar{c}_1, \dots, \bar{c}_n\}$
- pour chaque couleur  $c$  trouver la classe  $C_k$  qui la contient :

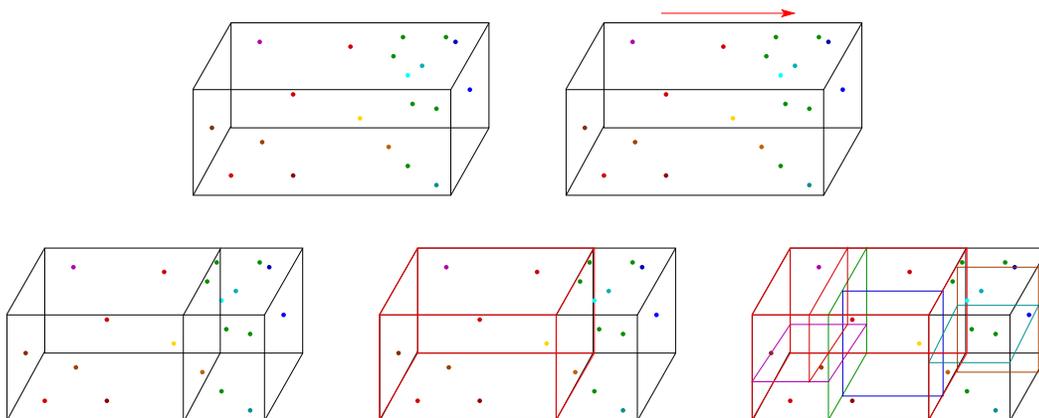
$$Q_I(c) = \bar{c}_k$$

de manière à minimiser  $EQ(I, Q_I(I))$

- mais problème **NP-complet** en dimension 3  $\rightarrow$  heuristiques
- différentes stratégies
  - méthodes de *popularité* : couleurs les plus présentes (résultats médiocres)
  - méthodes de *fusion*
  - méthodes de *division*
  - etc.

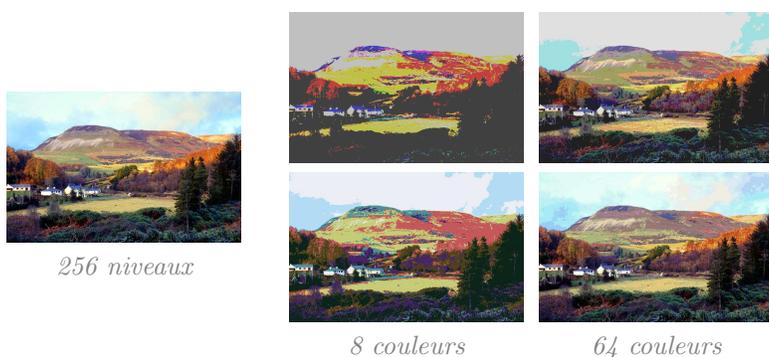
### Algorithme d'Heckbert : coupe médiane (ou median cut) – division [1982]

- construire une **boîte parallélépipédique RGB** minimale contenant toutes les couleurs de l'image
- sélectionner l'axe (R,G,B) le long duquel lequel la taille de la boîte est **maximale**
- couper la boîte au **point médian** (même nombre de points dans l'image pour chaque sous-ensemble de couleur)
- recommencer avec la boîte de **plus grande taille** jusqu'à obtenir le nombre de boîtes recherché



- représentant : **moyenne** des couleurs de la classe
- résultat codable par un **arbre binaire de recherche**
- nombreuses variantes : équilibrage des sommes de **variances**, espaces de couleur **uniformes**, découpe en plusieurs sous-boîtes, axes de composantes principales, etc.

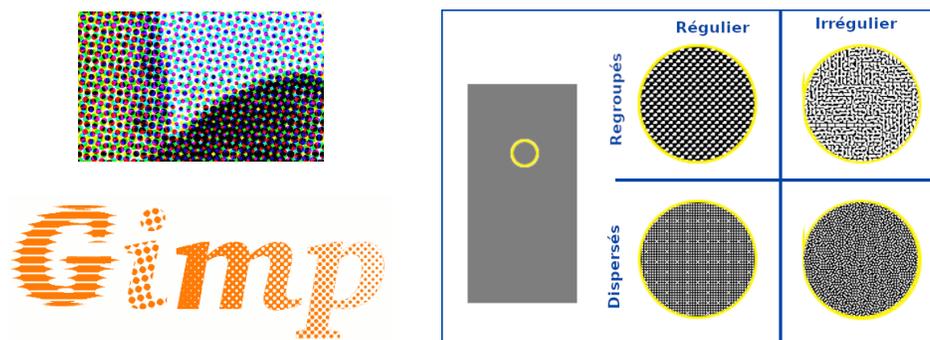
## Comparaison de quantification uniforme (ligne 1) et adaptative par coupe médiane (ligne 2)



## 5 Tramage (*dithering*)

### 5.1 Principe

- simuler une gamme de gris ou de couleurs par variation de densité de points (effet intégrateur de l'œil / mélange optique)
  - utilisation de technologie à rendu binaire (impression monochrome, quadrichromie, etc.)
  - compensation les défauts de quantification
  - effets graphiques



## 5.2 Tramage irrégulier : diffusion d'erreur

- Floyd et Steinberg (1976)
- principe de base : adaptation locale de la quantification uniforme sur deux niveaux

$$\begin{aligned} \text{si } (I[i][j] > \frac{1}{2}) \text{ alors } I'[i][j] &= 1 \\ \text{sinon } I'[i][j] &= 0 \end{aligned}$$

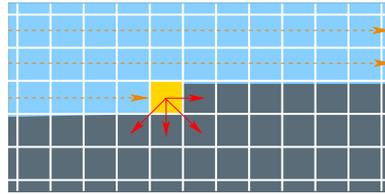
- pour chaque point seuillé, calcul de l'erreur commise  $e$

$$e = I'[i][j] - I[i][j]$$

exemple : si  $I(P) = 0.7$ , on a  $0.7 > 0.5 \rightarrow I'(p) = 1$  et  $e = 0.3$

- propagation de  $-e$  sur les 4 voisins non encore traités

$$e \times \begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \diamond & \frac{7}{16} \\ \frac{3}{16} & \frac{5}{16} & \frac{1}{16} \end{pmatrix}$$



- applicable à toute quantification : monochrome ou RGB, uniforme ou adaptative

### Exemple de tramage par diffusion d'erreur



Seuillage binaire    Seuillage binaire avec diffusion d'erreur

### Exemple de tramage / diffusion d'erreur associé à une quantification sur huit niveaux



256 niveaux de gris

uniforme

adaptative



quantification uniforme avec diffusion d'erreur



quantification adaptative avec diffusion d'erreur



I (162263 couleurs)



quantification uniforme de I sur huit couleurs (seuillage  $2 \times 2 \times 2$ ) avec diffusion d'erreur

## 6 Quantification des images à grande gamme dynamique (HDR)

### 6.1 Image à grande gamme dynamique



- synthèse d'image : adaptation locale de la quantification
- traitement local du contraste
- acquisition : combinaison de plusieurs photos

Question : à quand remonte la première acquisition HDR ?

### 6.2 Dynamique d'une image

- rend compte de l'étendue d'une plage de luminosité (ou d'intensité) conformément à la loi de Weber-Fechner :

$$S_{max}:\Delta_S \text{ avec } \begin{cases} S_{max} : \text{plus grand signal possible} \\ \Delta_S : \text{plus petit écart de signal} \end{cases}$$

- sensibilité de **réception** d'un signal : œil, capteur photo, etc
- capacité de **restitution** : écran
- mesure logarithmique :  $\log\left(\frac{S_{max}}{\Delta_S}\right)$ 
  - logarithme base 10 : dB (décibels)
  - logarithme base 2 : « diaphs » (diaphragme) ou « stops » ou EV (exposure value)
- **exemples** :
  - sensibilité de l'**œil humain** : jusqu'à  $2^{24}:1$  (24 stops)
  - dynamique d'une **image 8 bits** par canal :  $256:1$  (8 stops)
  - dynamique d'une photographie format **raw** : 12 à 16 stops

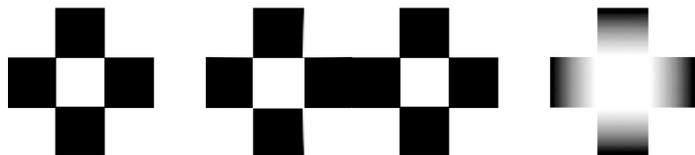
- exemples d'images de scènes à grande dynamique



pb d'affichage sur un écran d'une image avec une grande dynamique ( $> 8$  stops)

## 6.3 Quantification d'images HDR

- quantification locale : images au format raw, images de synthèse



Exemple d'effet de halo produit par variation locale de contraste [Zavagno and Caputo 2001]

- requantification par post-traitement : rehaussement de contraste local (C.2)



image de dynamique 256:1  
40784 couleurs

traitement 1  
154018 couleurs

traitement 2  
100496 couleurs

- combinaison d'acquisitions avec différents réglages d'exposition
  - *ouverture de diaphragme*
  - *sensibilité du signal mesuré par le capteur*
- construction d'une carte de luminance HDR à partir des expositions multiples permettant de pondérer la contribution de chaque image en chaque pixel



Jiang Duan and Guoping Qiu [2004]



Exemple avec un traitement standard de smartphone

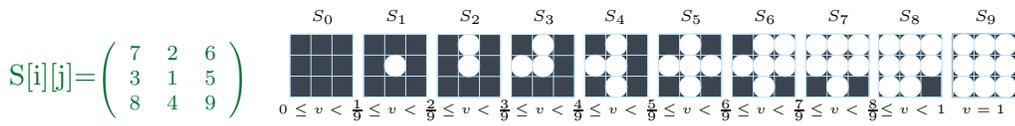
## ANNEXE A

### Tramage régulier (points regroupés ou dispersés)

#### Simulation de niveaux de gris par des motifs

- méthode naïve : chaque point est représenté par un carré  $p \times p$  pixels
- génération de motifs **imbriqués** : codables par une matrice de  $p^2$  entiers consécutifs

— exemple de matrice  $3 \times 3$  : 10 niveaux de gris

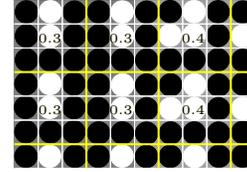


— on ne code qu'un point sur neuf : perte de résolution (grossissement des pixels)

exemple : deux rectangles gris d'intensité 0.3 et 0.4

$$v = 0.3 \quad \rightarrow \quad \frac{2}{9} < v < \frac{3}{9} \quad \rightarrow \quad S_2$$

$$v = 0.4 \quad \rightarrow \quad \frac{3}{9} < v < \frac{4}{9} \quad \rightarrow \quad S_3$$

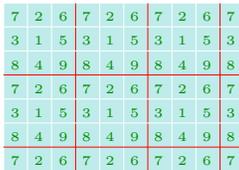


### Conservation de la résolution : méthode de seuillage

— utilisation de la matrice de motif  $S$  comme matrice de seuillage

— *pavage de l'image avec la matrice de motifs*

— *en chaque point, seuillage du niveau de gris par la valeur correspondante de  $S$*



$$0.3 > \frac{1}{9} = 0.111\dots 0.3 \not> \frac{4}{9} = 0.444\dots$$



- même résultat sur des plages uniformes
- pas de perte de résolution (seuillage par pixel)
- densité de niveau satisfaite **statistiquement**

$$\text{si } (I[i][j] \geq \frac{1}{9} * S[i \% 3][j \% 3])$$

$$I'[i][j] = 1$$

$$\text{sinon}$$

$$I'[i][j] = 0$$