Partie C — thème 1

Quantification d'image

256 niveaux

6 niveaux

Partie C — thème 1

Quantification d'image

162263 couleurs

8 couleurs

1. Introduction

1.1. Objectifs de la quantification

- discrétiser une information continue de luminance ou de couleurs :

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

- 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

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

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

1. Introduction

1.1. Objectifs de la quantification

- discrétiser une information continue de luminance ou de couleurs :

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

- 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

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

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

1. Introduction

1.1. Objectifs de la quantification

- discrétiser une information continue de luminance ou de couleurs :

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

- 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

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

  - *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 $\overline{c}_i$ pour chaque classe c_i*

exemple :

i	[0..10]	[11..68]	[69..83]	[83..142]	[143..255]
$\overline{c_i}$	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)

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 $\overline{c}_i$ pour chaque classe c_i*

exemple :

i	[0..10]	[11..68]	[69..83]	[83..142]	[143..255]
$\overline{c_i}$	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)

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 $\overline{c}_i$ pour chaque classe c_i*

exemple :

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

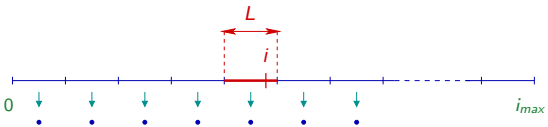
  - *fonction de quantification $Q(i)$: pour chaque pixel p, $I'(p) = \overline{c_{I(p)}}$*

- 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)

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

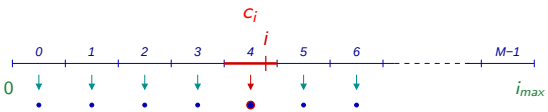
exemple $i_{max} = 255$ et $M = 10$; largeur : $L = 25.5$

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

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)

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

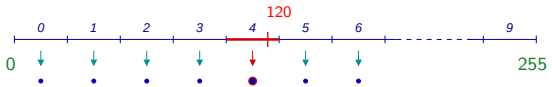
exemple $i_{max} = 255$ et $M = 10$; largeur : $L = 25.5$

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

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)

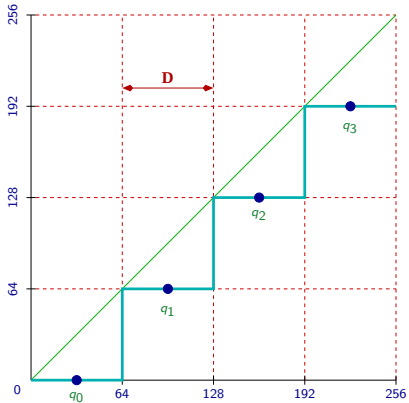
$$c_i = \left\lfloor \frac{i}{L} \right\rfloor$$

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

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

- choisir le représentant de la classe c_i

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

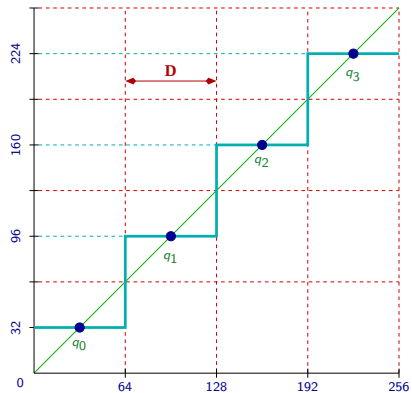


$$\begin{aligned} [0..63] &\rightarrow 32 \\ [64..127] &\rightarrow 96 \\ [128..191] &\rightarrow 160 \\ [192..255] &\rightarrow 224 \end{aligned}$$

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

- choisir le représentant de la classe c_i

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



$$\begin{aligned} [0..63] &\rightarrow 32 \\ [64..127] &\rightarrow 96 \\ [128..191] &\rightarrow 160 \\ [192..255] &\rightarrow 224 \end{aligned}$$

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

- Exemple :

256 niveaux

8 niveaux

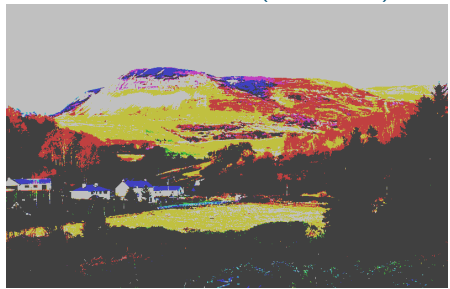
4 niveaux

○ Images couleur

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

256 niveaux

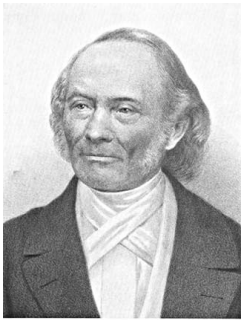
$4 \times 4 \times 4$ couleurs (39 utilisées)



$2 \times 2 \times 2$ couleurs

3. Loi de Weber-Fechner

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

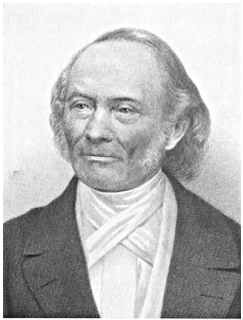


- loi formulée par Gustav Fechner (1801-1887) à partir des travaux expérimentaux d'Ernst Weber (1795-1878)
- étude quantitative du lien entre stimulus physiques et sensation (comparaisons de poids, de sons, de teintes)
- plus petite différence perceptible = fraction constante de l'excitant

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

3. Loi de Weber-Fechner

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

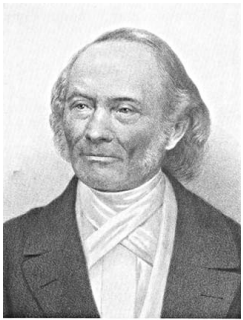


- loi formulée par Gustav Fechner (1801-1887) à partir des travaux expérimentaux d'Ernst Weber (1795-1878)
- étude quantitative du lien entre stimulus physiques et sensation (comparaisons de poids, de sons, de teintes)
- plus petite différence perceptible = fraction constante de l'excitant

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

3. Loi de Weber-Fechner

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

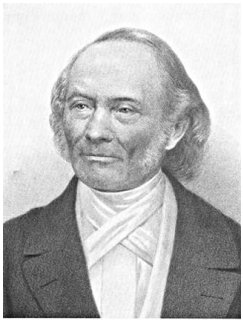


- loi formulée par Gustav Fechner (1801-1887) à partir des travaux expérimentaux d'Ernst Weber (1795-1878)
- étude quantitative du lien entre stimulus physiques et sensation (comparaisons de poids, de sons, de teintes)
- plus petite différence perceptible = fraction constante de l'excitant

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

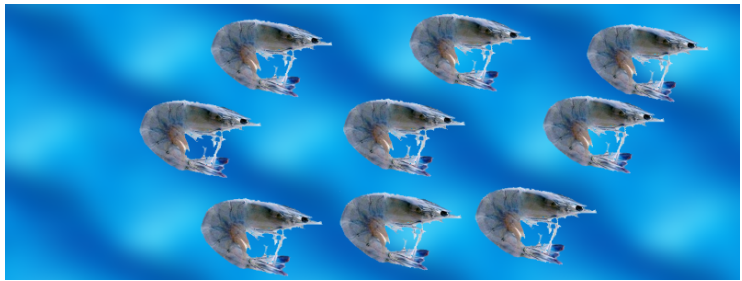
3. Loi de Weber-Fechner

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



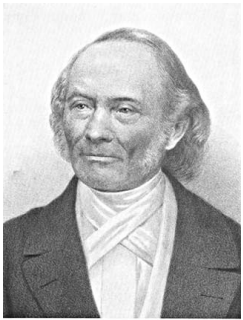
- loi formulée par Gustav Fechner (1801-1887) à partir des travaux expérimentaux d'Ernst Weber (1795-1878)
- étude quantitative du lien entre stimulus physiques et sensation (comparaisons de poids, de sons, de teintes)
- plus petite différence perceptible = fraction constante de l'excitant

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



3. Loi de Weber-Fechner

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

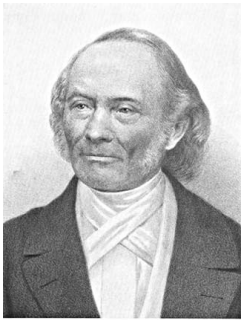


- loi formulée par Gustav Fechner (1801-1887) à partir des travaux expérimentaux d'Ernst Weber (1795-1878)
- étude quantitative du lien entre stimulus physiques et sensation (comparaisons de poids, de sons, de teintes)
- plus petite différence perceptible = fraction constante de l'excitant

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

3. Loi de Weber-Fechner

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

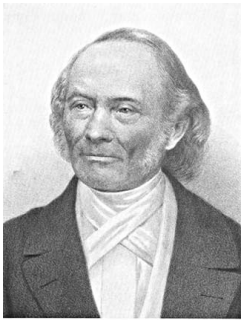


- loi formulée par Gustav Fechner (1801-1887) à partir des travaux expérimentaux d'Ernst Weber (1795-1878)
- étude quantitative du lien entre stimulus physiques et sensation (comparaisons de poids, de sons, de teintes)
- plus petite différence perceptible = fraction constante de l'excitant

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

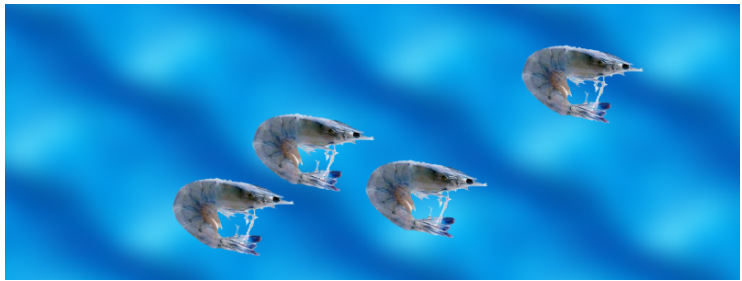
3. Loi de Weber-Fechner

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



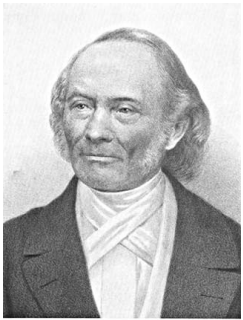
- loi formulée par Gustav Fechner (1801-1887) à partir des travaux expérimentaux d'Ernst Weber (1795-1878)
- étude quantitative du lien entre stimulus physiques et sensation (comparaisons de poids, de sons, de teintes)
- plus petite différence perceptible = fraction constante de l'excitant

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



3. Loi de Weber-Fechner

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

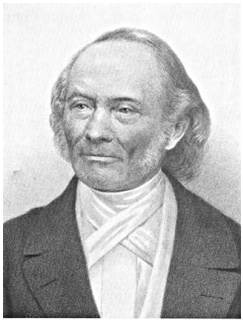


- loi formulée par Gustav Fechner (1801-1887) à partir des travaux expérimentaux d'Ernst Weber (1795-1878)
- étude quantitative du lien entre stimulus physiques et sensation (comparaisons de poids, de sons, de teintes)
- plus petite différence perceptible = fraction constante de l'excitant

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

3. Loi de Weber-Fechner

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



- 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

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

- **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)*

$\uparrow$

$\frac{1}{\alpha}$

$\downarrow$

$\simeq 3$ *(env. très clair)*

○ 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 :*

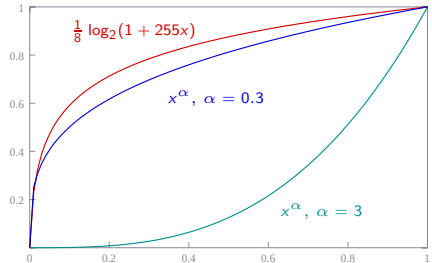
$\simeq \frac{1}{3}$ *(env. sombre)*

$\uparrow$

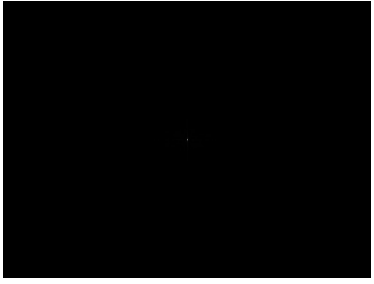
$\frac{1}{\alpha}$

$\downarrow$

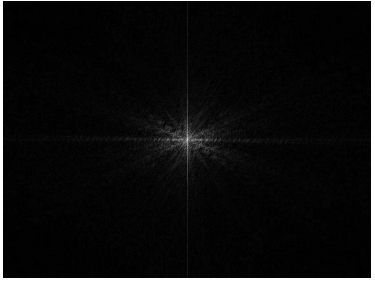
$\simeq 3$ *(env. très clair)*



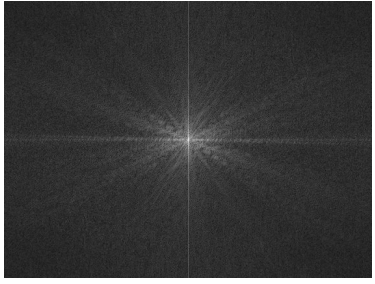
○ Exemple : visualisation de spectre d'amplitude



quantification uniforme
(dynamique $> 10^6$)



idem après correction logarithmique

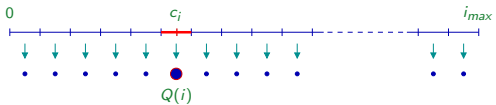


idem après correction $x^{0.2}$

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

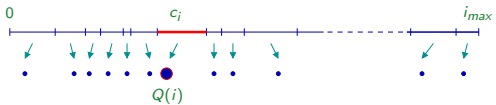


- objectif : minimiser l'écart entre I et $Q_I(I)$: différence quadratique

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

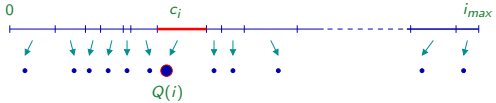


- objectif : minimiser l'écart entre I et $Q_I(I)$: différence quadratique

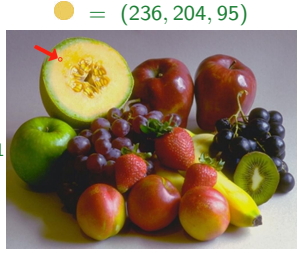
4. Quantification adaptative

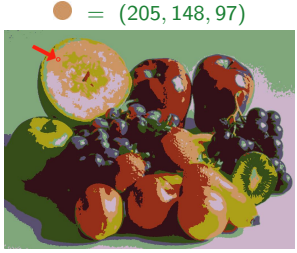
4.1. Quantification optimale

- 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





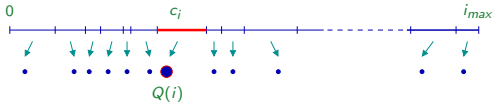
$$\begin{aligned} & (236-205)^2 \\ + \; & (204-148)^2 \\ + \; & (95-97)^2 \\ = \; & 4101 \end{aligned}$$

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

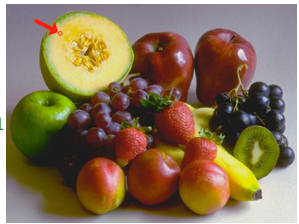
4. Quantification adaptative

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



I_1

I_2

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

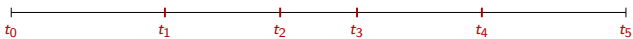
$$DQ(I_1, I_2) = \sum_{p \in D} (I_1(p) - I_2(p))^2$$

4.2. Images monochromes : quantification sous-optimale

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

$$\begin{gathered}[t_0\,,\,t_1] \\ [t_1\,,\,t_2] \\ \vdots \\ [t_{M-1}\,,\,t_M] \\ \text{avec} \quad t_0 = 0 < t_1 < \ldots < t_M = N-1\end{gathered}$$

- choix d'un représentant r_j pour chaque intervalle



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

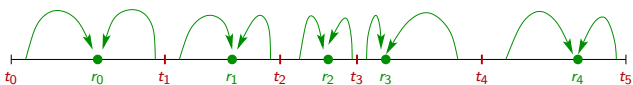
- trouver les $\{t_j\}$ et $\{r_j\}$ minimisant la différence quadratique entre I et $Q(I)$

4.2. Images monochromes : quantification sous-optimale

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

$$\begin{gathered}[t_0\,,\,t_1]\\ [t_1\,,\,t_2]\\ \vdots\\ [t_{M-1}\,,\,t_M]\end{gathered}$$
$$\text{avec} \quad t_0 = 0 < t_1 < \ldots < t_M = N-1$$

- choix d'un représentant r_j pour chaque intervalle



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

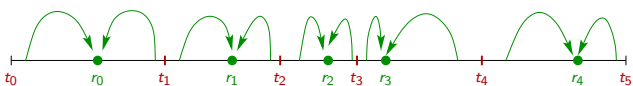
- trouver les $\{t_j\}$ et $\{r_j\}$ minimisant la différence quadratique entre I et $Q(I)$

4.2. Images monochromes : quantification sous-optimale

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

$$\begin{gathered}[t_0\,,\,t_1] \\ [t_1\,,\,t_2] \\ \vdots \\ [t_{M-1}\,,\,t_M] \\ \text{avec} \quad t_0 = 0 < t_1 < \ldots < t_M = N-1\end{gathered}$$

- choix d'un représentant r_j pour chaque intervalle



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

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

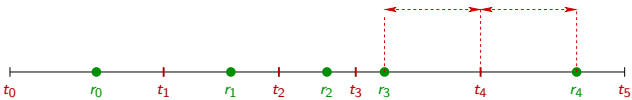
- on peut montrer que les $\{t_j\}$ et $\{r_j\}$ doivent satisfaire deux conditions :
  - t_i est le milieu du segment $\overline{r_{i-1} r_i}$

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

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

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

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



- conditions nécessaires mais non suffisantes : sous-optimal

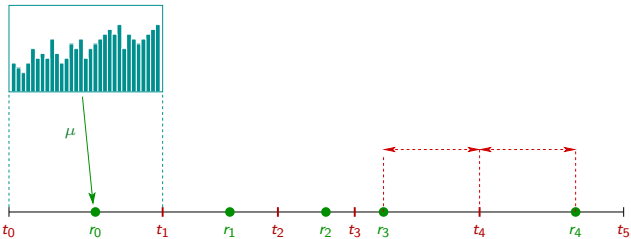
- on peut montrer que les $\{t_j\}$ et $\{r_j\}$ doivent satisfaire deux conditions :
  - t_i est le milieu du segment $\overline{r_{i-1} r_i}$

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

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

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

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



- conditions nécessaires mais non suffisantes : sous-optimal

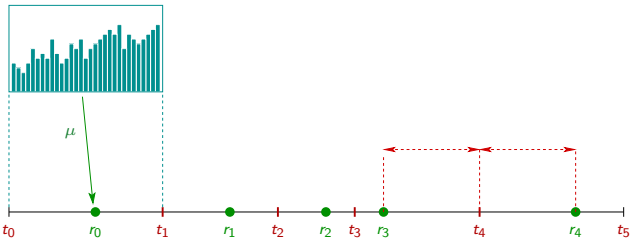
- on peut montrer que les $\{t_j\}$ et $\{r_j\}$ doivent satisfaire deux conditions :
  - t_i est le milieu du segment $\overline{r_{i-1} r_i}$

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

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

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

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



- conditions nécessaires mais non suffisantes : sous-optimal

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie

(1)

- 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

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie

(2)

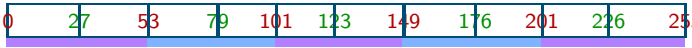
- algorithme de Lloyd-Max : répéter
  1. calcul des r_i en fonction des t_i (moyennes)
  2. calcul des t_i en fonction des r_i (milieux)

  jusqu'à la stabilité des données
- solution sous-optimale mais convergence rapide

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie



(3)

- 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

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie

(4)

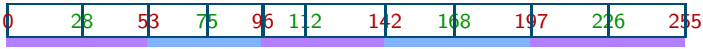
- algorithme de Lloyd-Max : répéter
  1. calcul des r_i en fonction des t_i (moyennes)
  2. calcul des t_i en fonction des r_i (milieux)

  jusqu'à la stabilité des données
- solution sous-optimale mais convergence rapide

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie



(5)

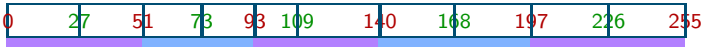
- algorithme de Lloyd-Max : répéter
  1. calcul des r_i en fonction des t_i (moyennes)
  2. calcul des t_i en fonction des r_i (milieux)

  jusqu'à la stabilité des données
- solution sous-optimale mais convergence rapide

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie



(6)

- 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

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie

(7)

- 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

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie

(8)

- 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

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie

(9)

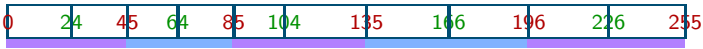
- algorithme de Lloyd-Max : répéter
  1. calcul des r_i en fonction des t_i (moyennes)
  2. calcul des t_i en fonction des r_i (milieux)

  jusqu'à la stabilité des données
- solution sous-optimale mais convergence rapide

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie



(10)

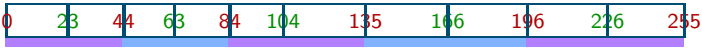
- algorithme de Lloyd-Max : répéter
  1. calcul des r_i en fonction des t_i (moyennes)
  2. calcul des t_i en fonction des r_i (milieux)

  jusqu'à la stabilité des données
- solution sous-optimale mais convergence rapide

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie



(11)

- 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

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie

(12)

- 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

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie

(13)

- 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

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie

(14)

- 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

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie

(15)

- 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

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie

(16)

- 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

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie

(17)

- 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

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

$\xrightarrow{5}$

- initialisation des t_i de manière équirépartie

(18)

- 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

uniforme

5 niveaux

8 niveaux

adaptative

○ Exemples

256 niveaux

uniforme

5 niveaux

8 niveaux

adaptative

○ Exemples

256 niveaux

uniforme

5 niveaux

8 niveaux

adaptative

★ Quantification d'images en niveaux de gris (1)

I_s

I

J

K

Les images I, J et K montrent les résultats de trois quantifications sur six niveaux de gris de l'image de gauche. Retrouvez pour chaque image la méthode de quantification utilisée.

$\alpha = 1.6 \quad (\frac{1}{\alpha} = 0.625)$	A	B	C	D
uniforme	K	J	J	I
uniforme après correction $I_s^{1/\alpha}$	I	K	I	K
adaptative	J	I	K	J

★ Quantification d'images en niveaux de gris (1)

I_s

I

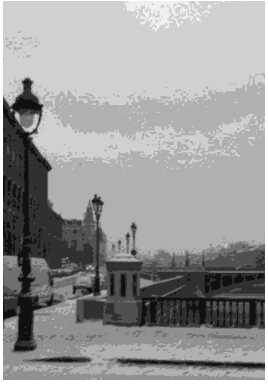
J

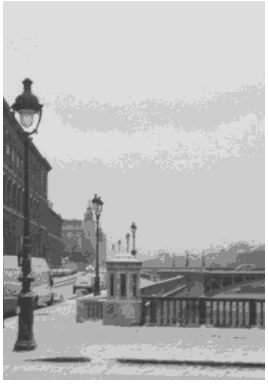
K

Les images I, J et K montrent les résultats de trois quantifications sur six niveaux de gris de l'image de gauche. Retrouvez pour chaque image la méthode de quantification utilisée.

$\alpha = 1.6 \quad (\frac{1}{\alpha} = 0.625)$	**A**	**B**	**C**	**D**
uniforme	K	J	J	I
uniforme après correction $I_s^{1/\alpha}$	I	K	I	K
adaptative	J	I	K	J

★ Quantification d'images en niveaux de gris (2)





I J K

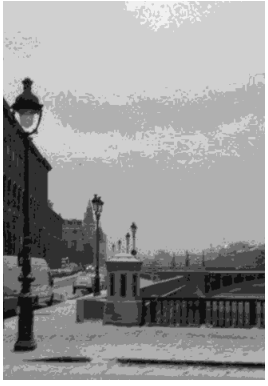
Les images I, J et K montrent les résultats de trois quantifications sur huit niveaux de gris de l'image de gauche. Retrouvez pour chaque image la méthode de quantification utilisée.

$\alpha = 0.6 \quad (\frac{1}{\alpha} = 1.666\ldots)$

	A	B	C	D
uniforme	I	K	K	J
uniforme après correction $I'^{1/\alpha}$	J	J	I	I
adaptative	K	I	J	K

★ Quantification d'images en niveaux de gris (2)

I



J

K

Les images I, J et K montrent les résultats de trois quantifications sur huit niveaux de gris de l'image de gauche. Retrouvez pour chaque image la méthode de quantification utilisée.

$\alpha = 0.6 \quad (\frac{1}{\alpha} = 1.666\ldots)$	**A**	**B**	**C**	**D**
uniforme	I	K	K	J
uniforme après correction $I'^{1/\alpha}$	J	J	I	I
adaptative	K	I	J	K

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*

$$C = C_1 \cup \ldots \cup C_n \quad et \quad C_i \cap C_j = \varnothing$$

  - *choisir les représentants des classes :* $\{\bar{c}_1, \ldots, \bar{c}_n\}$
  - *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 $\longrightarrow$ heuristiques
- différentes stratégies
  - *méthodes de popularité : couleurs les plus présentes (résultats médiocres)*
  - *méthodes de fusion*
  - *méthodes de division*
  - *etc.*

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*

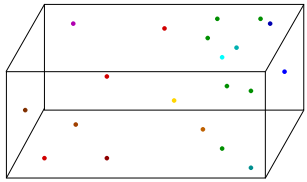
$$C = C_1 \cup \ldots \cup C_n \quad et \quad C_i \cap C_j = \varnothing$$

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

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

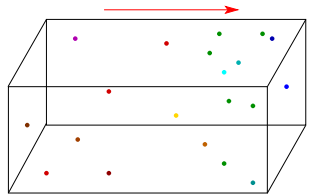
  de manière à minimiser $EQ(I, Q_I(I))$
- mais problème NP-complet en dimension 3 $\longrightarrow$ heuristiques
- différentes stratégies
  - *méthodes de popularité : couleurs les plus présentes (résultats médiocres)*
  - *méthodes de fusion*
  - *méthodes de division*
  - *etc.*

- **Algorithme d'Heckbert : coupe médiane (ou median cut) – division [1982]**
  - construire une boîte parallélépipèdique 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



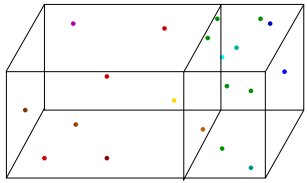
  - représentant : moyenne des couleurs de la classe
  - résultat codable par un arbre binaire de recherche
  - nombreuses variantes : équilibrage des sommes de variances, espaces de couleur uniformes, découpe en plusieurs sous-boîtes, axes de composantes principales, etc.

- **Algorithme d'Heckbert : coupe médiane (ou median cut) – division [1982]**
  - construire une boîte parallélépipèdique 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



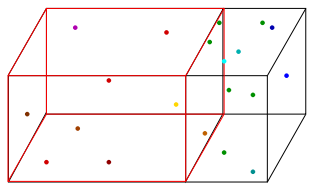
  - représentant : moyenne des couleurs de la classe
  - résultat codable par un arbre binaire de recherche
  - nombreuses variantes : équilibrage des sommes de variances, espaces de couleur uniformes, découpe en plusieurs sous-boîtes, axes de composantes principales, etc.

- **Algorithme d'Heckbert : coupe médiane (ou median cut) – division [1982]**
  - construire une boîte parallélépipèdique 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



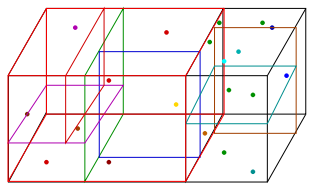
  - représentant : moyenne des couleurs de la classe
  - résultat codable par un arbre binaire de recherche
  - nombreuses variantes : équilibrage des sommes de variances, espaces de couleur uniformes, découpe en plusieurs sous-boîtes, axes de composantes principales, etc.

- **Algorithme d'Heckbert : coupe médiane (ou median cut) – division [1982]**
  - construire une boîte parallélépipèdique 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



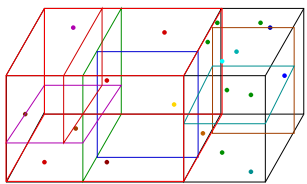
  - représentant : moyenne des couleurs de la classe
  - résultat codable par un arbre binaire de recherche
  - nombreuses variantes : équilibrage des sommes de variances, espaces de couleur uniformes, découpe en plusieurs sous-boîtes, axes de composantes principales, etc.

- **Algorithme d'Heckbert : coupe médiane (ou median cut) – division [1982]**
  - construire une boîte parallélépipèdique 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.

- **Algorithme d'Heckbert : coupe médiane (ou median cut) – division [1982]**
  - construire une boîte parallélépipèdique 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)**

256 niveaux

8 couleurs

64 couleurs

★ Quantification optimale

Certaines des méthodes de quantification suivantes sont elles construites sur un principe de recherche de solution optimale ?

	A	B	C	D	E
quantification uniforme	non	non	non	non	oui
algorithme de Lloyd-Max	non	oui	non	oui	oui
algorithme d'Heckbert	non	non	oui	oui	oui

★ Quantification optimale

Certaines des méthodes de quantification suivantes sont elles construites sur un principe de recherche de solution optimale ?

	A	B	C	D	E
quantification uniforme	non	non	non	non	oui
algorithme de Lloyd-Max	non	oui	non	oui	oui
algorithme d'Heckbert	non	non	oui	oui	oui

Une méthode optimale

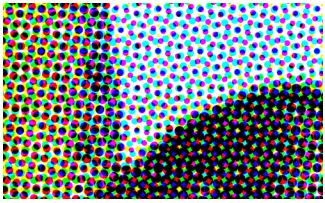
— dépend de l'image : ce n'est pas le cas de la quantification uniforme

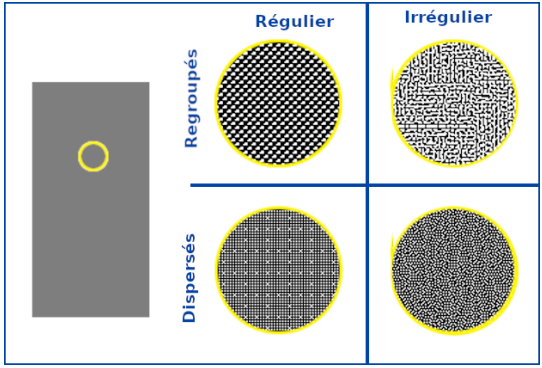
— optimise un objectif (ici minimiser $EQ(I, Q_I(I))$) : ce n'est pas le cas de la coupe médiane qui est une heuristique

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 monchrome, 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

$$\begin{array}{l} \texttt{si } (I[i][j] > \frac{1}{2}) \texttt{ alors I'[i][j] = 1} \\ \qquad\qquad\qquad\quad \texttt{sinon I'[i][j] = 0} \end{array}$$

- 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 $0.7 > 0.5 \longrightarrow I'(p) = 1$ 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

5.2. Tramage irrégulier : diffusion d'erreur

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

$$\begin{array}{l} \texttt{si } (I[i][j] > \frac{1}{2}) \texttt{ alors } I'[i][j] = 1 \\ \qquad\qquad\qquad\quad \texttt{sinon } I'[i][j] = 0 \end{array}$$

- 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 $0.7 > 0.5 \longrightarrow I'(p) = 1$ 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

5.2. Tramage irrégulier : diffusion d'erreur

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

```
si (I[i][j] > 1/2) alors I'[i][j] = 1
                   sinon I'[i][j] = 0
```

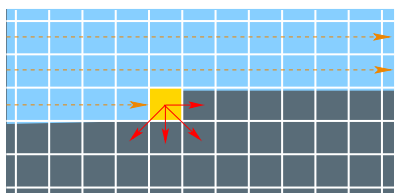
- pour chaque point seuillé, calcul de l'erreur commise e

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

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

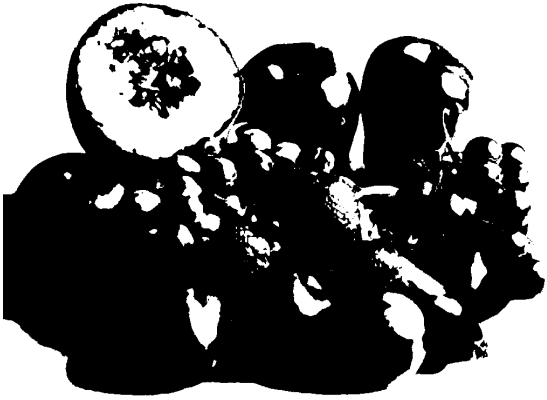
- propagation de $-e$ sur les 4 voisins non encore traités

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



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

○ Exemple de tramage par diffusion d'erreur

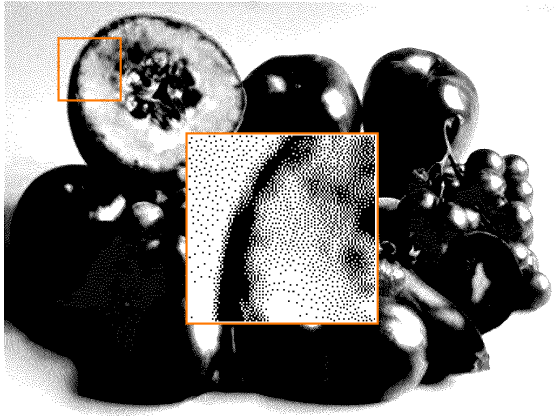


Seuillage binaire

Seuillage binaire avec diffusion d'erreur

○ Exemple de tramage par diffusion d'erreur

Seuillage binaire



Seuillage binaire avec diffusion d'erreur

○ Exemple de tramage / diffusion d'erreur associé à une quantificationsur huit niveaux

256 niveaux de gris

quantification sur huit niveaux de gris

uniforme

adaptative

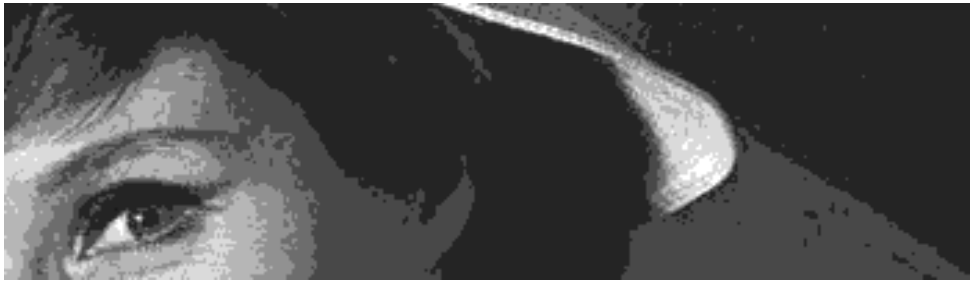
○ Exemple de tramage / diffusion d'erreur associé à une quantificationsur huit niveaux

256 niveaux de gris

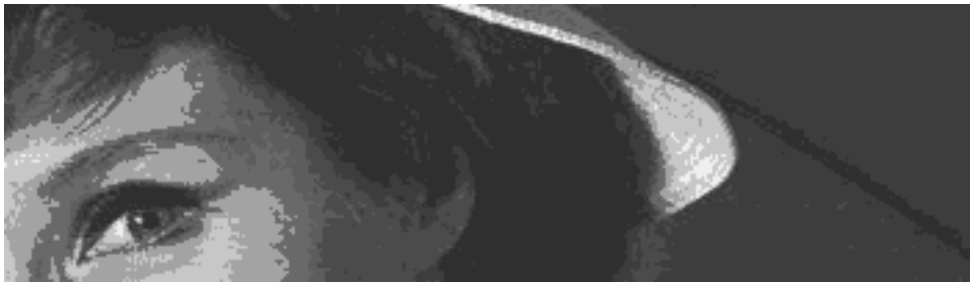
quantification sur huit niveaux de gris

uniforme

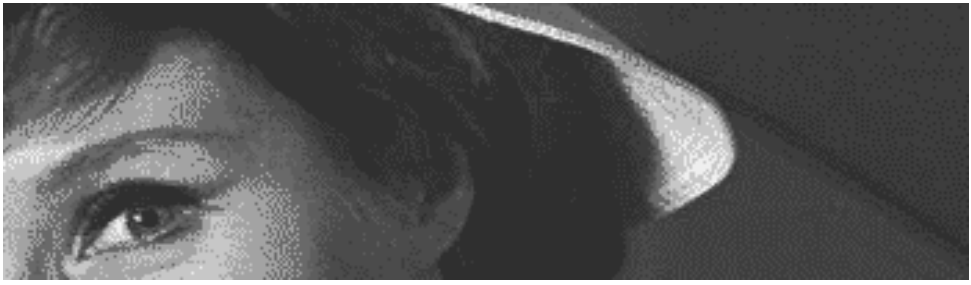
adaptative



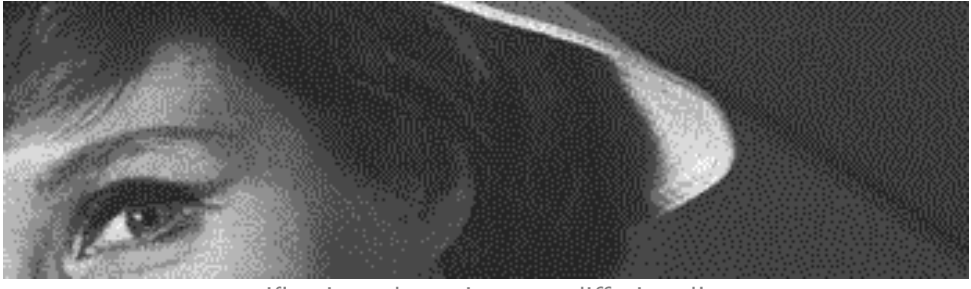
quantification uniforme



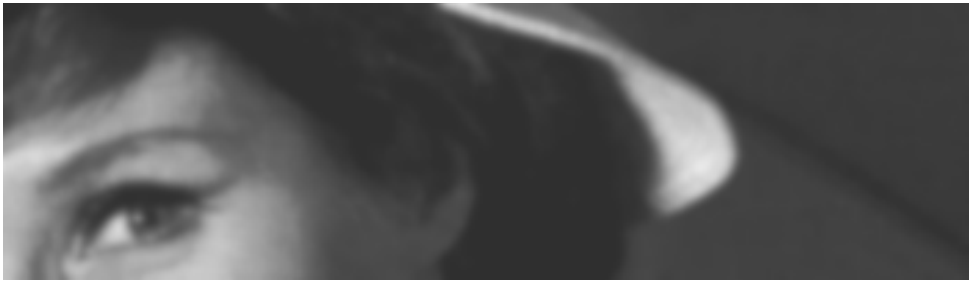
quantification adaptative



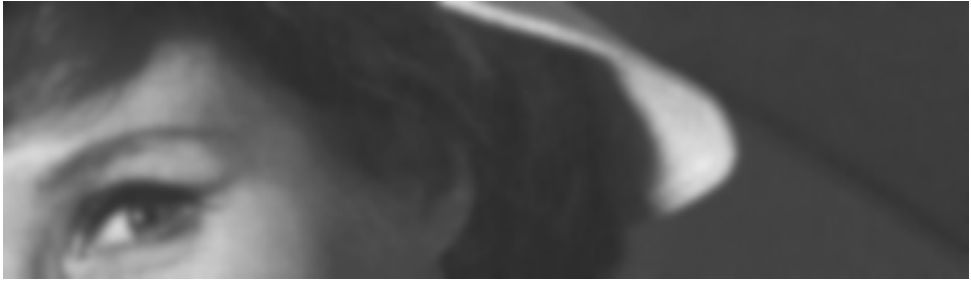
quantification uniforme avec diffusion d'erreur



quantification adaptative avec diffusion d'erreur



quantification uniforme avec diffusion d'erreur



quantification adaptative avec diffusion d'erreur

I (162263 couleurs)

quantification uniforme de I sur huit couleurs (seuillage $2 \times 2 \times 2$)

I (162263 couleurs)

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

I (162263 couleurs)

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

★ Écart entre deux images

I : image RGB en 55764 couleurs

A : quantification CM de I en 256 couleurs

B : idem avec tramage par diffusion d'erreur

différence quadratique : $DQ(I, I') = \sum_{P \in D} (I_R(P) - I'_R(P))^2 + (I_G(P) - I'_G(P))^2 + (I_B(P) - I'_B(P))^2$

A : $DQ(I, A) > DQ(I, B)$ **B** : $DQ(I, A) < DQ(I, B)$

★ Écart entre deux images

I

★ Écart entre deux images

A : CM 128 couleurs

★ Écart entre deux images

B : CM 128 couleurs
diffusion d'erreur

★ Écart entre deux images

B : CM 128 couleurs
diffusion d'erreur

★ Écart entre deux images

I : image RGB en 55764 couleurs

A : quantification CM de I en 256 couleurs

B : idem avec tramage par diffusion d'erreur

différence quadratique : $DQ(I, I') = \sum_{P \in D} (I_R(P) - I'_R(P))^2 + (I_G(P) - I'_G(P))^2 + (I_B(P) - I'_B(P))^2$

A : $DQ(I, A) > DQ(I, B)$ **B** : $DQ(I, A) < DQ(I, B)$

★ Écart entre deux images

I : image RGB en 55764 couleurs

A : quantification CM de I en 256 couleurs

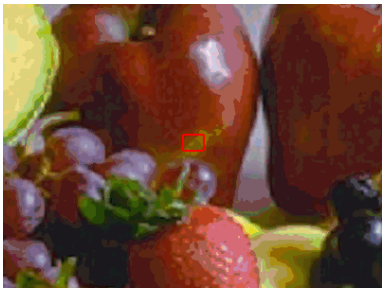
B : idem avec tramage par diffusion d'erreur

différence quadratique : $DQ(I, I') = \sum_{P \in D} (I_R(P) - I'_R(P))^2 + (I_G(P) - I'_G(P))^2 + (I_B(P) - I'_B(P))^2$

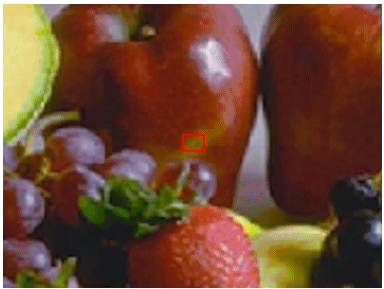
A : $DQ(I, A) > DQ(I, B)$ **B** : $DQ(I, A) < DQ(I, B)$

★ Écart entre deux images

I



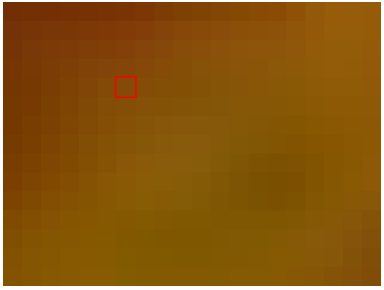
A : sans diffusion d'erreur



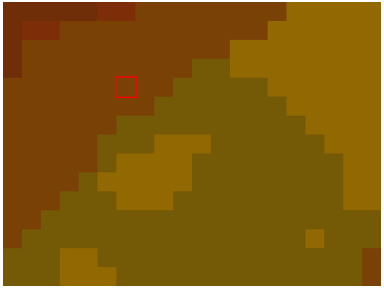
B : avec diffusion d'erreur

$$(I(p) - A(p))^2 < (I(p) - B(p))^2$$

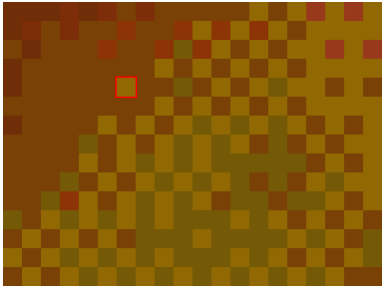
★ Écart entre deux images



I



A : sans diffusion d'erreur



B : avec diffusion d'erreur

$$(I(p) - A(p))^2 < (I(p) - B(p))^2$$

6. Quantification d'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. Quantification d'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 :

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

  - 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

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 :

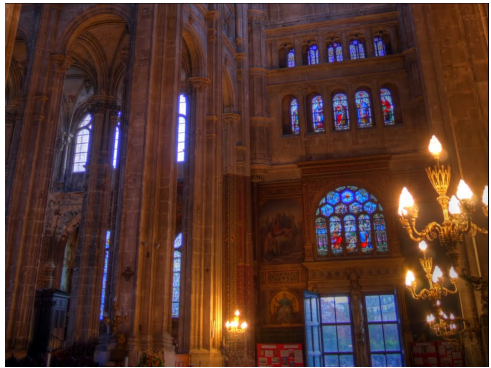
$$S_{max}{:}\Delta_S \quad \text{avec} \quad \begin{cases} S_{max} : \text{plus grand signal possible} \\ \Delta_S : \text{plus petit écart de signal} \end{cases}$$

  - 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

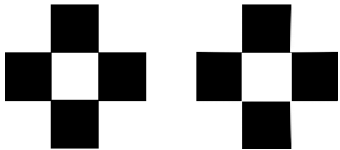
- exemples d'images de scènes à grande dynamique



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

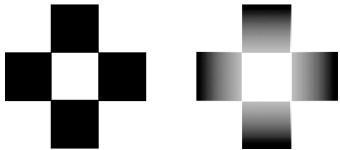


Exemple d'effet de halo produit par variation locale de contraste [Zavagno and Caputo 2001]

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

6.3. Quantification d'images HDR

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

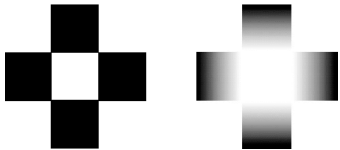


Exemple d'effet de halo produit par variation locale de contraste [Zavagno and Caputo 2001]

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

6.3. Quantification d'images HDR

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

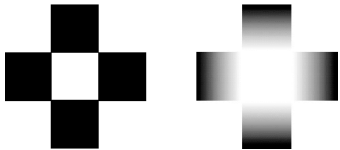


Exemple d'effet de halo produit par variation locale de contraste [Zavagno and Caputo 2001]

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

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

la technique d'*acquisitions multiples* a été mise au point par Gustave le Gray, vers 1856 : tirage séparé de négatifs du paysage et du ciel (environ 30 ans après l'invention de la photographie par Joseph Niépce).

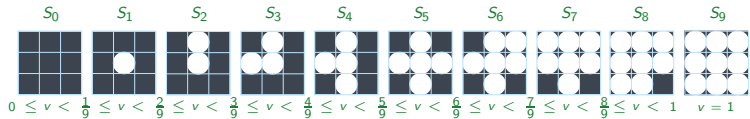
ANNEXE A

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 3×3 : 10 niveaux de gris

$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



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

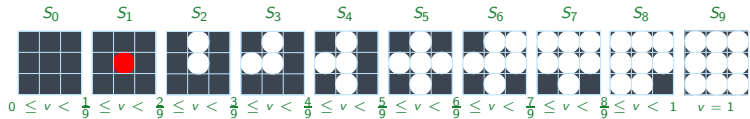
exemple : deux rectangles gris d'intensité 0.3 et 0.4

$$v = 0.3 \longrightarrow \frac{2}{9} < v < \frac{3}{9} \longrightarrow S_2$$
$$v = 0.4 \longrightarrow \frac{3}{9} < v < \frac{4}{9} \longrightarrow S_3$$

○ 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 3×3 : 10 niveaux de gris

$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



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

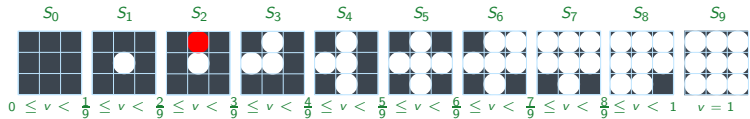
exemple : deux rectangles gris d'intensité 0.3 et 0.4

$$v = 0.3 \longrightarrow \frac{2}{9} < v < \frac{3}{9} \longrightarrow S_2$$
$$v = 0.4 \longrightarrow \frac{3}{9} < v < \frac{4}{9} \longrightarrow S_3$$

○ 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 3×3 : 10 niveaux de gris

$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



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

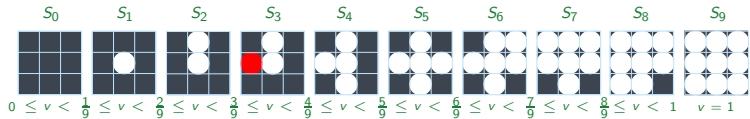
exemple : deux rectangles gris d'intensité 0.3 et 0.4

$$v = 0.3 \longrightarrow \frac{2}{9} < v < \frac{3}{9} \longrightarrow S_2$$
$$v = 0.4 \longrightarrow \frac{3}{9} < v < \frac{4}{9} \longrightarrow S_3$$

○ 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 3×3 : 10 niveaux de gris

$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



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

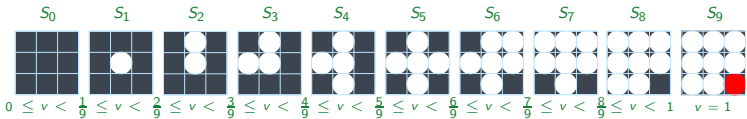
exemple : deux rectangles gris d'intensité 0.3 et 0.4

$$v = 0.3 \longrightarrow \frac{2}{9} < v < \frac{3}{9} \longrightarrow S_2$$
$$v = 0.4 \longrightarrow \frac{3}{9} < v < \frac{4}{9} \longrightarrow S_3$$

○ 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 3×3 : 10 niveaux de gris

$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



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

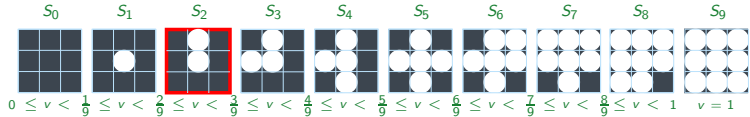
exemple : deux rectangles gris d'intensité 0.3 et 0.4

$$v = 0.3 \longrightarrow \frac{2}{9} < v < \frac{3}{9} \longrightarrow S_2$$
$$v = 0.4 \longrightarrow \frac{3}{9} < v < \frac{4}{9} \longrightarrow S_3$$

○ 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 3×3 : 10 niveaux de gris

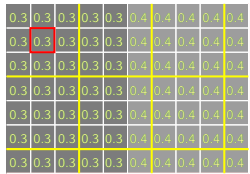
$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



- 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

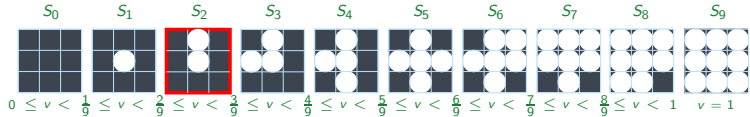
$$v = 0.3 \longrightarrow \frac{2}{9} < v < \frac{3}{9} \longrightarrow S_2$$
$$v = 0.4 \longrightarrow \frac{3}{9} < v < \frac{4}{9} \longrightarrow S_3$$



○ 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 3×3 : 10 niveaux de gris

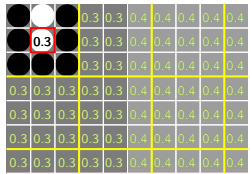
$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



- 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

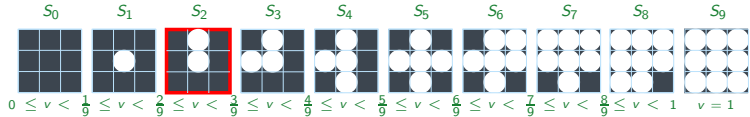
$$v = 0.3 \longrightarrow \frac{2}{9} < v < \frac{3}{9} \longrightarrow S_2$$
$$v = 0.4 \longrightarrow \frac{3}{9} < v < \frac{4}{9} \longrightarrow S_3$$



○ 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 3×3 : 10 niveaux de gris

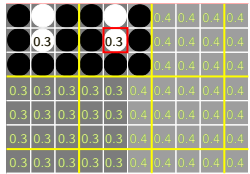
$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



- 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

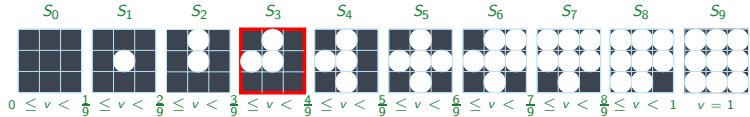
$$v = 0.3 \longrightarrow \frac{2}{9} < v < \frac{3}{9} \longrightarrow S_2$$
$$v = 0.4 \longrightarrow \frac{3}{9} < v < \frac{4}{9} \longrightarrow S_3$$



○ 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 3×3 : 10 niveaux de gris

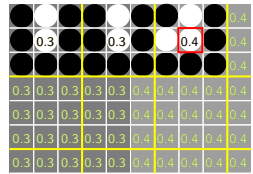
$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



- 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

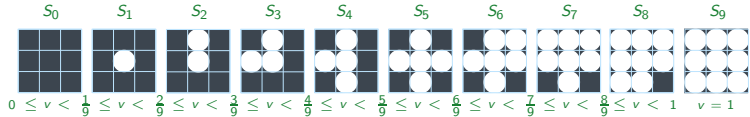
$$v = 0.3 \longrightarrow \frac{2}{9} < v < \frac{3}{9} \longrightarrow S_2$$
$$v = 0.4 \longrightarrow \frac{3}{9} < v < \frac{4}{9} \longrightarrow S_3$$



○ 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 3×3 : 10 niveaux de gris

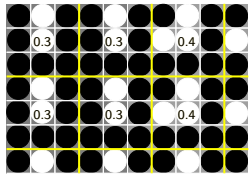
$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



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

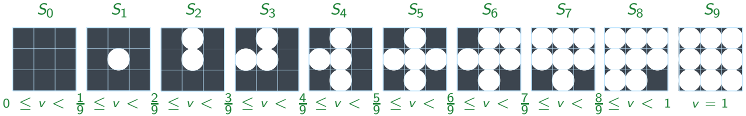
exemple : deux rectangles gris d'intensité 0.3 et 0.4

$$v = 0.3 \longrightarrow \frac{2}{9} < v < \frac{3}{9} \longrightarrow S_2$$
$$v = 0.4 \longrightarrow \frac{3}{9} < v < \frac{4}{9} \longrightarrow S_3$$



○ Conservation de la résolution : méthode de seuillage

$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



- 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 conrrespondante de* S

7	2	6	7	2	6	7	2	6	7
3	1	5	3	1	5	3	1	5	3
8	4	9	8	4	9	8	4	9	8
7	2	6	7	2	6	7	2	6	7
3	1	5	3	1	5	3	1	5	3
8	4	9	8	4	9	8	4	9	8
7	2	6	7	2	6	7	2	6	7

$$0.3 > \frac{1}{9} = 0.111...$$

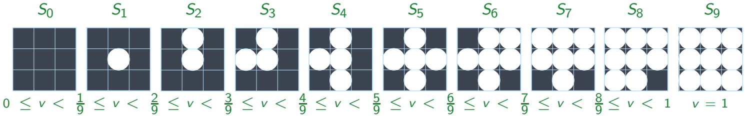
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4

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

si $(I[i][j] \geq \frac{1}{9} * S[i \% 3][j \% 3])$
```
    I'[i][j] = 1
sinon
    I'[i][j] = 0
```

○ Conservation de la résolution : méthode de seuillage

$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



- 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 conrrespondante de* S

7	2	6	7	2	6	7	2	6	7
3	1	5	3	1	5	3	1	5	3
8	4	9	8	4	9	8	4	9	8
7	2	6	7	2	6	7	2	6	7
3	1	5	3	1	5	3	1	5	3
8	4	9	8	4	9	8	4	9	8
7	2	6	7	2	6	7	2	6	7

$$0.3 > \frac{1}{9} = 0.111...$$

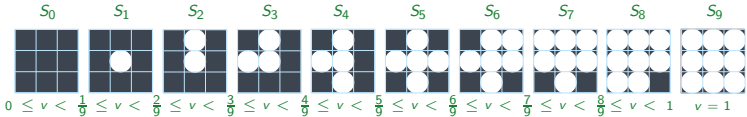
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4
0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.4

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

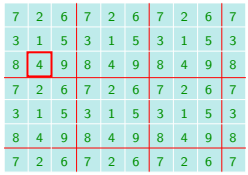
```
si (I[i][j] ≥ 1/9 * S[i % 3][j % 3])
    I'[i][j] = 1
sinon
    I'[i][j] = 0
```

○ Conservation de la résolution : méthode de seuillage

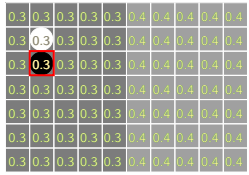
$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



- 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 conrrespondante de S*



$$0.3 \not> \frac{4}{9} = 0.444...$$

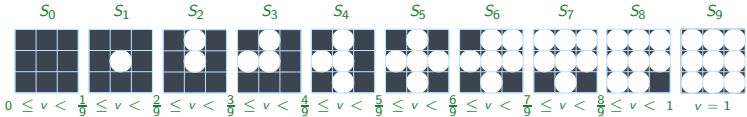


- même résultat sur des plages uniformes
- pas de perte de résolution (seuillage par pixel)
- densité de niveau satisfaite statistiquement

```
si (I[i][j] ≥ 1/9 * S[i % 3][j % 3])
    I'[i][j] = 1
sinon
    I'[i][j] = 0
```

○ Conservation de la résolution : méthode de seuillage

$$S[i][j] = \begin{pmatrix} 7 & 2 & 6 \\ 3 & 1 & 5 \\ 8 & 4 & 9 \end{pmatrix}$$



- 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 conrrespondante de S*

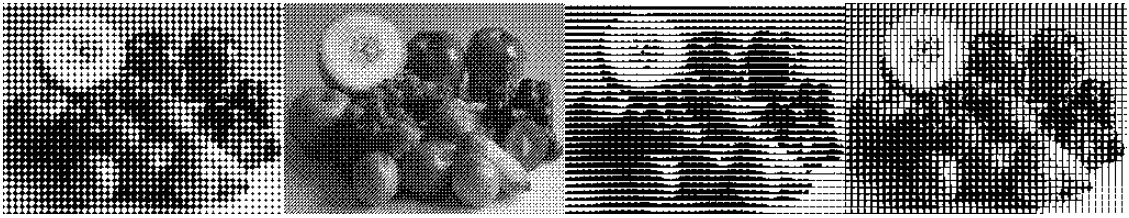
7	2	6	7	2	6	7	2	6	7
3	1	5	3	1	5	3	1	5	3
8	4	9	8	4	9	8	4	9	8
7	2	6	7	2	6	7	2	6	7
3	1	5	3	1	5	3	1	5	3
8	4	9	8	4	9	8	4	9	8
7	2	6	7	2	6	7	2	6	7

$0.3 \not> \frac{4}{9} = 0.444...$

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

```
si (I[i][j] ≥ 1/9 * S[i % 3][j % 3])
    I'[i][j] = 1
sinon
    I'[i][j] = 0
```

★ Tramage



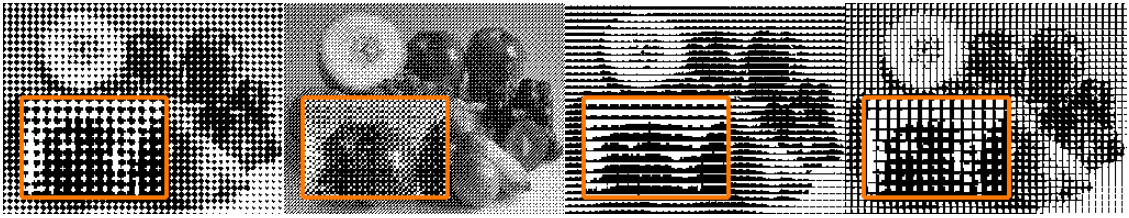
I_1 I_2 I_3 I_4

$$D_1 = \begin{pmatrix} 21 & 20 & 19 & 18 & 17 & 36 \\ 22 & 7 & 6 & 5 & 16 & 35 \\ 23 & 8 & 1 & 4 & 15 & 34 \\ 24 & 9 & 2 & 3 & 14 & 33 \\ 25 & 10 & 11 & 12 & 13 & 32 \\ 26 & 27 & 28 & 29 & 30 & 31 \end{pmatrix} \quad D_2 = \begin{pmatrix} 26 & 16 & 7 & 15 & 25 & 32 \\ 17 & 8 & 2 & 6 & 14 & 24 \\ 9 & 3 & 1 & 5 & 13 & 23 \\ 18 & 10 & 4 & 12 & 22 & 31 \\ 27 & 19 & 11 & 21 & 30 & 35 \\ 33 & 28 & 20 & 29 & 34 & 36 \end{pmatrix} \quad D_3 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 \\ 19 & 20 & 21 & 22 & 23 & 24 \\ 25 & 26 & 27 & 28 & 29 & 30 \\ 31 & 32 & 33 & 34 & 35 & 36 \end{pmatrix} \quad D_4 = \begin{pmatrix} 1 & 29 & 13 & 3 & 31 & 15 \\ 25 & 21 & 9 & 27 & 23 & 11 \\ 17 & 5 & 33 & 19 & 7 & 35 \\ 4 & 32 & 16 & 2 & 30 & 14 \\ 28 & 24 & 12 & 26 & 22 & 10 \\ 20 & 8 & 36 & 18 & 6 & 34 \end{pmatrix}$$

	I_1	I_2	I_3	I_4
A	D_1	D_4	D_3	D_2
B	D_2	D_4	D_3	D_1

	I_1	I_2	I_3	I_4
C	D_2	D_3	D_4	D_1
D	D_2	D_4	D_1	D_3

★ Tramage



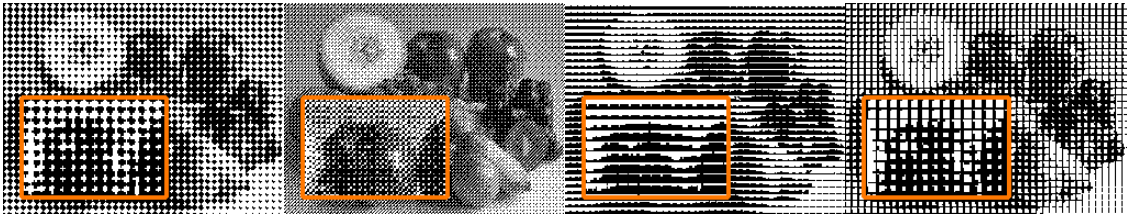
I_1 I_2 I_3 I_4

$$D_1 = \begin{pmatrix} 21 & 20 & 19 & 18 & 17 & 36 \\ 22 & 7 & 6 & 5 & 16 & 35 \\ 23 & 8 & 1 & 4 & 15 & 34 \\ 24 & 9 & 2 & 3 & 14 & 33 \\ 25 & 10 & 11 & 12 & 13 & 32 \\ 26 & 27 & 28 & 29 & 30 & 31 \end{pmatrix} \quad D_2 = \begin{pmatrix} 26 & 16 & 7 & 15 & 25 & 32 \\ 17 & 8 & 2 & 6 & 14 & 24 \\ 9 & 3 & 1 & 5 & 13 & 23 \\ 18 & 10 & 4 & 12 & 22 & 31 \\ 27 & 19 & 11 & 21 & 30 & 35 \\ 33 & 28 & 20 & 29 & 34 & 36 \end{pmatrix} \quad D_3 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 \\ 19 & 20 & 21 & 22 & 23 & 24 \\ 25 & 26 & 27 & 28 & 29 & 30 \\ 31 & 32 & 33 & 34 & 35 & 36 \end{pmatrix} \quad D_4 = \begin{pmatrix} 1 & 29 & 13 & 3 & 31 & 15 \\ 25 & 21 & 9 & 27 & 23 & 11 \\ 17 & 5 & 33 & 19 & 7 & 35 \\ 4 & 32 & 16 & 2 & 30 & 14 \\ 28 & 24 & 12 & 26 & 22 & 10 \\ 20 & 8 & 36 & 18 & 6 & 34 \end{pmatrix}$$

	I_1	I_2	I_3	I_4
A	D_1	D_4	D_3	D_2
B	D_2	D_4	D_3	D_1

	I_1	I_2	I_3	I_4
C	D_2	D_3	D_4	D_1
D	D_2	D_4	D_1	D_3

★ Tramage



I_1 I_2 I_3 I_4

$$D_1 = \begin{pmatrix} 21 & 20 & 19 & 18 & 17 & 36 \\ 22 & 7 & 6 & 5 & 16 & 35 \\ 23 & 8 & 1 & 4 & 15 & 34 \\ 24 & 9 & 2 & 3 & 14 & 33 \\ 25 & 10 & 11 & 12 & 13 & 32 \\ 26 & 27 & 28 & 29 & 30 & 31 \end{pmatrix} \quad D_2 = \begin{pmatrix} 26 & 16 & 7 & 15 & 25 & 32 \\ 17 & 8 & 2 & 6 & 14 & 24 \\ 9 & 3 & 1 & 5 & 13 & 23 \\ 18 & 10 & 4 & 12 & 22 & 31 \\ 27 & 19 & 11 & 21 & 30 & 35 \\ 33 & 28 & 20 & 29 & 34 & 36 \end{pmatrix} \quad D_3 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 \\ 19 & 20 & 21 & 22 & 23 & 24 \\ 25 & 26 & 27 & 28 & 29 & 30 \\ 31 & 32 & 33 & 34 & 35 & 36 \end{pmatrix} \quad D_4 = \begin{pmatrix} 1 & 29 & 13 & 3 & 31 & 15 \\ 25 & 21 & 9 & 27 & 23 & 11 \\ 17 & 5 & 33 & 19 & 7 & 35 \\ 4 & 32 & 16 & 2 & 30 & 14 \\ 28 & 24 & 12 & 26 & 22 & 10 \\ 20 & 8 & 36 & 18 & 6 & 34 \end{pmatrix}$$

	I_1	I_2	I_3	I_4
A	D_1	D_4	D_3	D_2
B	D_2	D_4	D_3	D_1

	I_1	I_2	I_3	I_4
C	D_2	D_3	D_4	D_1
D	D_2	D_4	D_1	D_3