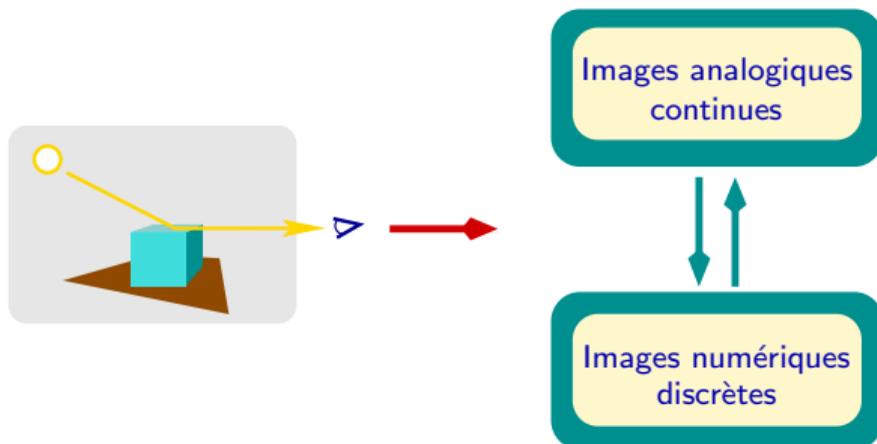


Partie A — thème 1

Formation et représentation spatiale des images 2D



1. De la lumière aux images

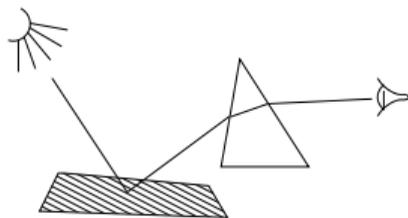
1.1. Interaction lumière / matière

- ▶ nature **corpusculaire** de la lumière : **rayons** lumineux

- ▶ *optique géométrique*
- ▶ *perception des formes*

- ▶ nature **ondulatoire** de la lumière : **ondes** lumineuses

- ▶ *phénomène vibratoire : longueur d'onde / fréquence*
- ▶ *perception des couleurs*
- ▶ *spectre visible : intervalle des longueurs d'onde perçues par l'œil humain*
- ▶ *distribution spectrale d'énergie (DSE)*

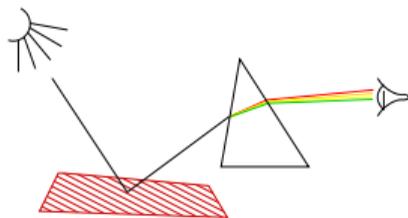


- ▶ **dualité onde-particule** : manifestation à notre échelle d'un phénomène quantique

1. De la lumière aux images

1.1. Interaction lumière / matière

- ▶ nature **corpusculaire** de la lumière : **rayons** lumineux
 - ▶ *optique géométrique*
 - ▶ *perception des formes*
- ▶ nature **ondulatoire** de la lumière : **ondes** lumineuses
 - ▶ *phénomène vibratoire : longueur d'onde / fréquence*
 - ▶ *perception des couleurs*
 - ▶ *spectre visible : intervalle des longueurs d'onde perçues par l'œil humain*
 - ▶ *distribution spectrale d'énergie (DSE)*



- ▶ **dualité onde-particule** : manifestation à notre échelle d'un phénomène quantique

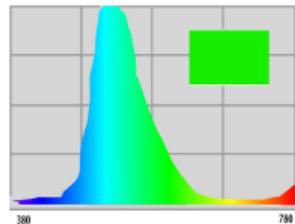
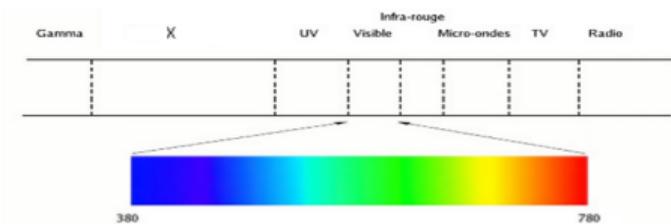
1. De la lumière aux images

1.1. Interaction lumière / matière

- ▶ nature **corpusculaire** de la lumière : **rayons** lumineux
 - ▶ *optique géométrique*
 - ▶ *perception des formes*
- ▶ nature **ondulatoire** de la lumière : **ondes** lumineuses
 - ▶ *phénomène vibratoire : longueur d'onde / fréquence*
 - ▶ *perception des couleurs*
 - ▶ **spectre visible** : *intervalle des longueurs d'onde perçues par l'œil humain*
 - ▶ **distribution spectrale d'énergie (DSE)**



$[\lambda_{min}, \lambda_{max}]$



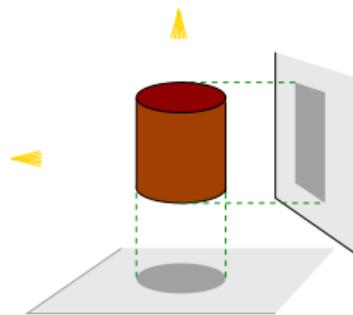
$E(\lambda)$

- ▶ **dualité onde-particule** : manifestation à notre échelle d'un phénomène quantique

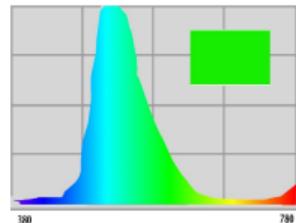
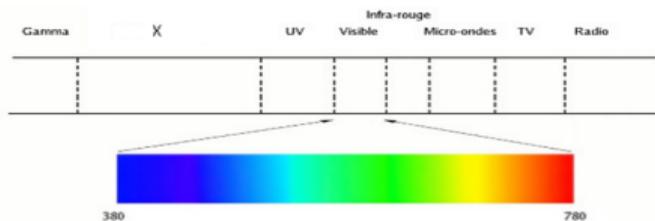
1. De la lumière aux images

1.1. Interaction lumière / matière

- ▶ nature **corpusculaire** de la lumière : **rayons** lumineux
 - ▶ *optique géométrique*
 - ▶ *perception des formes*
- ▶ nature **ondulatoire** de la lumière : **ondes** lumineuses
 - ▶ *phénomène vibratoire : longueur d'onde / fréquence*
 - ▶ *perception des couleurs*
 - ▶ **spectre visible** : *intervalle des longueurs d'onde perçues par l'œil humain*
 - ▶ **distribution spectrale d'énergie (DSE)**



$[\lambda_{min}, \lambda_{max}]$



$E(\lambda)$

- ▶ **dualité onde-particule** : manifestation à notre échelle d'un phénomène quantique

1.2. Qu'est ce qu'une image « naturelle » ?

- ▶ une image résulte d'informations lumineuses issues d'interactions lumière / matière

source lumineuse } interactions projection 2D
scène 3D → plan

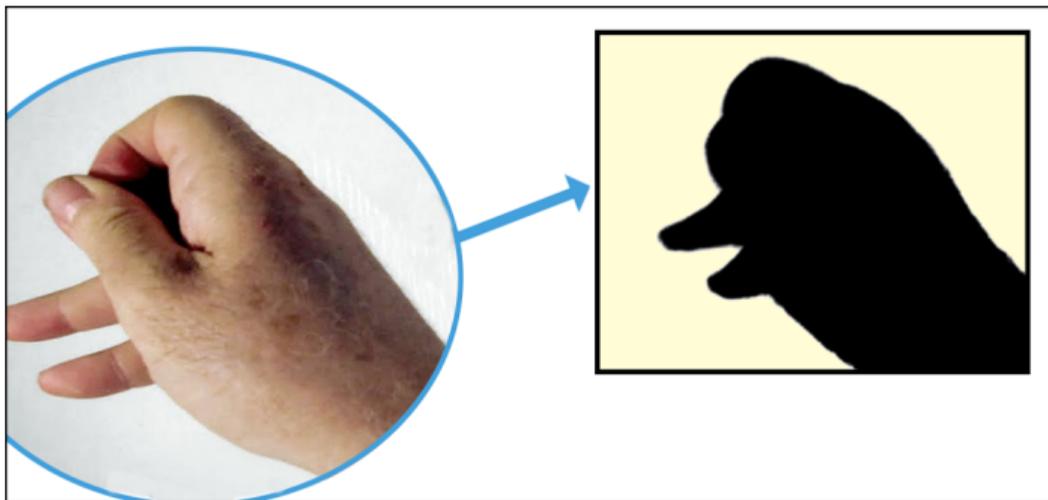


- ▶ une image dépend des propriétés du système de capture de l'information lumineuse

1.2. Qu'est ce qu'une image « naturelle » ?

- ▶ une image résulte d'informations lumineuses issues d'interactions lumière / matière

source lumineuse } interactions projection 2D
scène 3D → plan



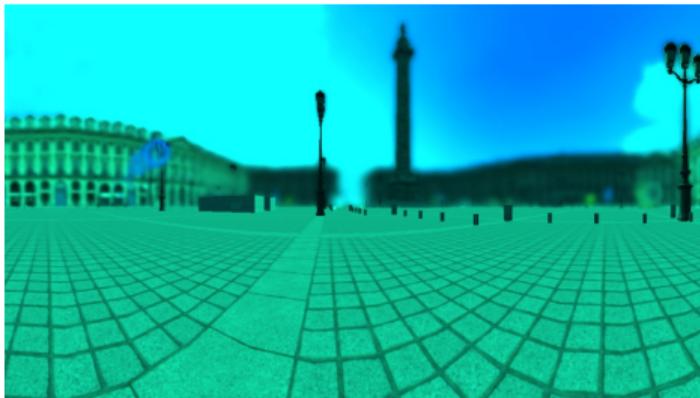
- ▶ une image dépend des propriétés du système de capture de l'information lumineuse

- ▶ exemple d'une scène urbaine vue par un système de vision humain moyen

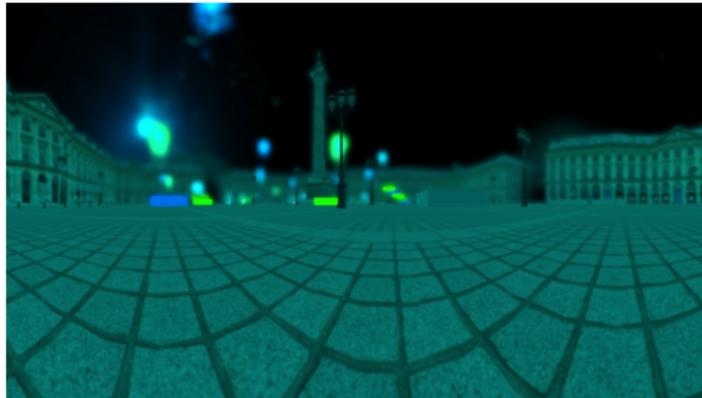


© 3DEXPERIENCE - Dassault Systèmes

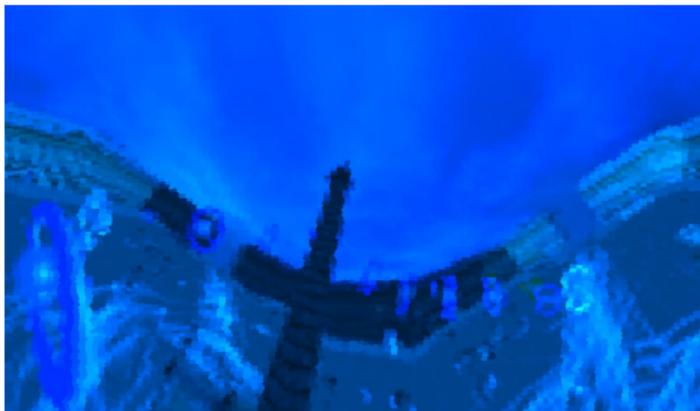
- ▶ cette même scène produit des images différentes avec d'autres systèmes de vision



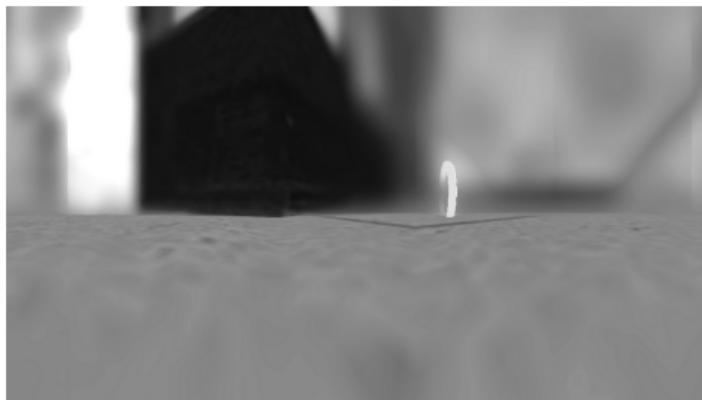
– vue par un chien –



– vue par un chat –



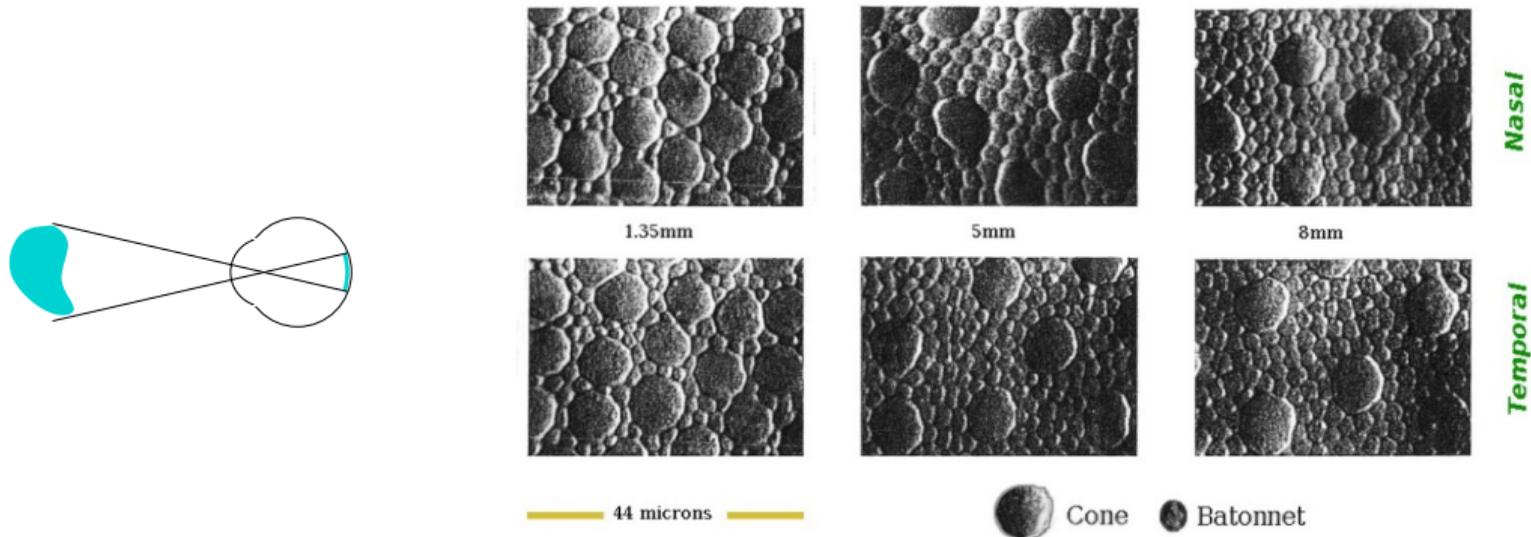
– vue par une abeille –



– vue par un rat –

1.3. Formation des images visuelles dites « naturelles »

- ▶ images acquises et construites par le **système de vision humain** (SVH)
- ▶ **projection** optique (lentille) : formation de l'image sur la rétine



- ▶ capteurs de la rétine (photorécepteurs) : mise en œuvre de deux processus
 - **échantillonnage** des points de l'image (position géométrique des points)
 - **mesure de la DSE** (couleur des points)

1.4. Des images pas si naturelles que cela...

○ L'antiquité grecque

- ▶ rayon de l'œil vers l'objet (pythagoriciens vers -550) vs rayon de l'objet vers l'œil (atomistes, Epicure vers -300)
- ▶ **Socrate** vers -450 puis **Platon** (vers -400) : interactions entre des émissions de l'objet et de l'œil
- ▶ **Aristote** (vers -360) : le jaune résulte d'un rapprochement entre le rouge et le vert
- ▶ **Euclide** (vers -300) : optique géométrique (rayons lumineux)
- ▶ **Ptolémée** (entre 100 et 150) : principes de réflexion/réfraction entre l'air et l'eau

○ L'âge d'or arabe

- ▶ **Al-Khwarizmi** (780 -850) : *Algoritmi*, « père de l'algèbre »
- ▶ **Al-Kindi** (vers 801-873) : toute chose dans le monde émet des rayons dans toutes les directions
- ▶ **Ibn Sahl** (vers 940-1000) : loi de la réfraction
- ▶ **Alhazenm, Ibn Al-Haytham** (traité d'optique 1015 - 1021) : « père de l'optique », première argumentation fondée en faveur de l'œil récepteur et non émetteur

1.4. Des images pas si naturelles que cela...

○ L'antiquité grecque

- ▶ rayon de l'œil vers l'objet (pythagoriciens vers -550) vs rayon de l'objet vers l'œil (atomistes, Epicure vers -300)
- ▶ Socrate vers -450 puis Platon (vers -400) : interactions entre des émissions de l'objet et de l'œil
- ▶ Aristote (vers -360) : le jaune résulte d'un rapprochement entre le rouge et le vert
- ▶ Euclide (vers -300) : optique géométrique (rayons lumineux)
- ▶ Ptolémée (entre 100 et 150) : principes de réflexion/réfraction entre l'air et l'eau

○ L'âge d'or arabe

- ▶ Al-Khwarizmi (780 -850) : *Algoritmi*, « père de l'algèbre »
- ▶ Al-Kindi (vers 801-873) : toute chose dans le monde émet des rayons dans toutes les directions
- ▶ Ibn Sahl (vers 940-1000) : loi de la réfraction
- ▶ Alhazenm, Ibn Al-Haytham (traité d'optique 1015 - 1021) : « père de l'optique », première argumentation fondée en faveur de l'œil récepteur et non émetteur

- **La Renaissance**

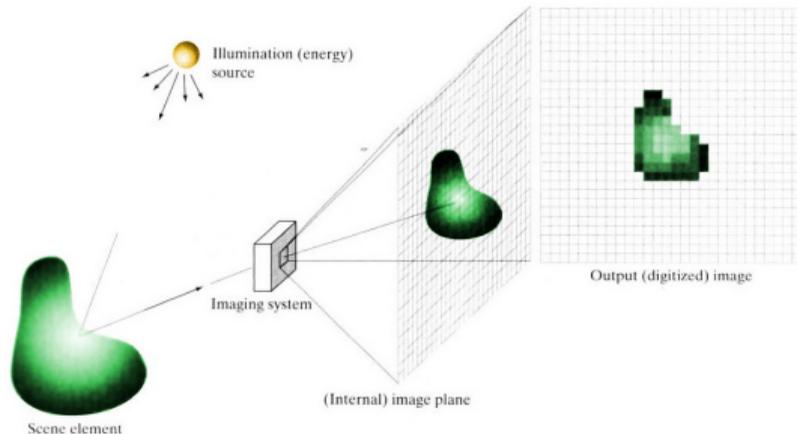
- ▶ **Alberti, Piero della Francesca** (à partir de 1430) : perspective

- **L'époque moderne**

- ▶ **Newton** (1642-1725) : théorie corpusculaire, cercle chromatique
- ▶ **Huygens** (1690) : modèle ondulatoire (Fresnel – 1815)
- ▶ **Helmholtz** (1821-1894) : trois types de récepteurs dans l'oeil (théorie des trois couleurs de Young-Helmholtz)
- ▶ **Chevreul** (1786-1889) : loi du contraste simultané (1839)
- ▶ **Maxwell** (1831-1879) : théorie électromagnétique
- ▶ **Planck** (1858-1947) : l'énergie est quantique
- ▶ **CIE, 1931** : courbes colorimétriques normalisées
- ▶ **Land, 1964..1987** : retinex, constance des couleurs

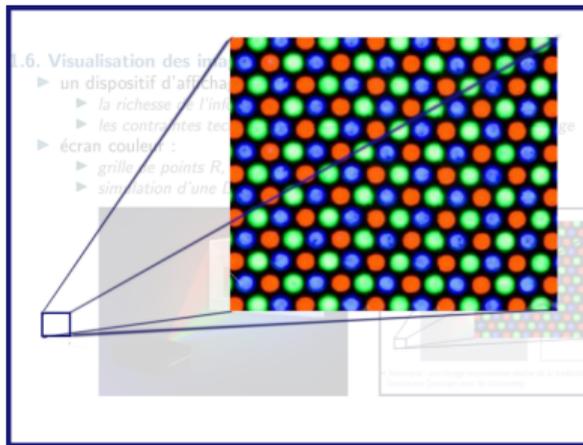
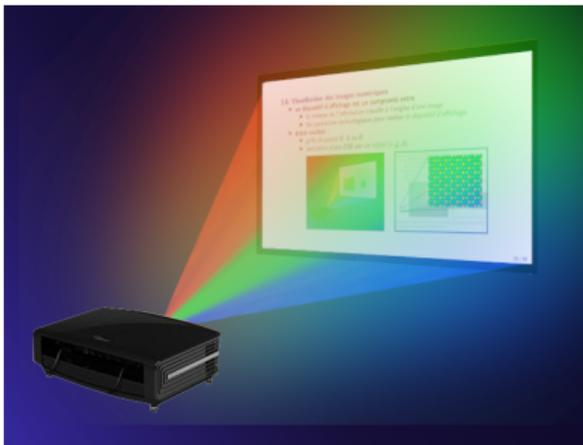
1.5. Acquisition des images numériques « naturelles »

- ▶ reproduction des propriétés du **système de vision humain (SVH)** : système optique analogue à l'œil humain
- ▶ projection 2D + acquisition dans le **spectre visible**
 - ▶ *lentille / rétine*
 - ▶ *trois types de cones* → *trois types capteurs R,G,B*

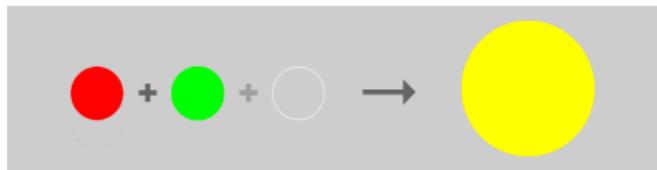


1.6. Visualisation des images numériques

- ▶ un dispositif d'affichage est un compromis entre
 - ▶ *la richesse de l'information visuelle à l'origine d'une image*
 - ▶ *les contraintes technologiques pour réaliser le dispositif d'affichage*
- ▶ simulation des DSE par des triplets de sources lumineuses (r, g, b)
 - ▶ *vidéoprojecteur : LCD, micromiroirs, etc.*
 - ▶ *écran : LCD, LED, technologie plasma, etc.*



(100%, 100%, 0%)



(100%, 50%, 0%)



(100%, 0%, 100%)

(50%, 0%, 100%)

(50%, 50%, 100%)

(100%, 100%, 0%)



(100%, 50%, 0%)



(100%, 0%, 100%)



(50%, 0%, 100%)



(50%, 50%, 100%)

(100%, 100%, 0%)



(100%, 50%, 0%)



(100%, 0%, 100%)



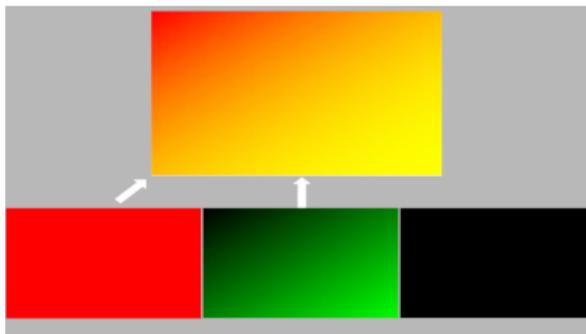
(50%, 0%, 100%)



(50%, 50%, 100%)

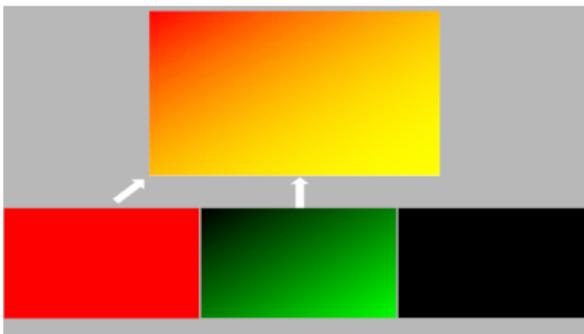


- Canaux R, G et B d'une image couleur

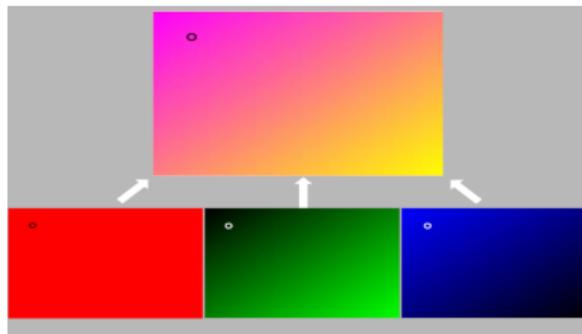


*Ex.1 : dégradé du rouge au
jaune (canal bleu nul)*

- Canaux R, G et B d'une image couleur

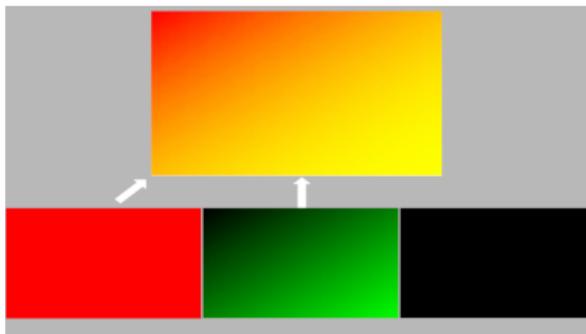


Ex.1 : dégradé du rouge au jaune (canal bleu nul)

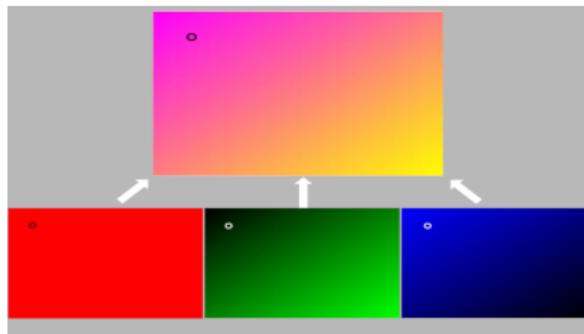


Ex.2 : idem avec dégradé du canal bleu

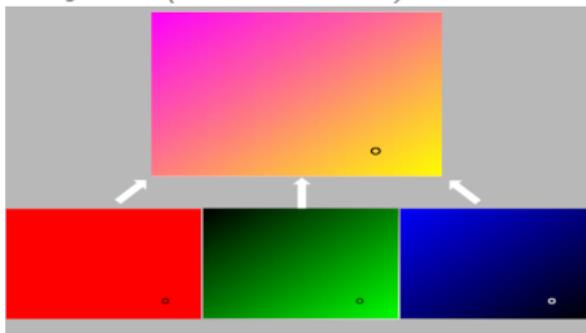
○ Canaux R, G et B d'une image couleur



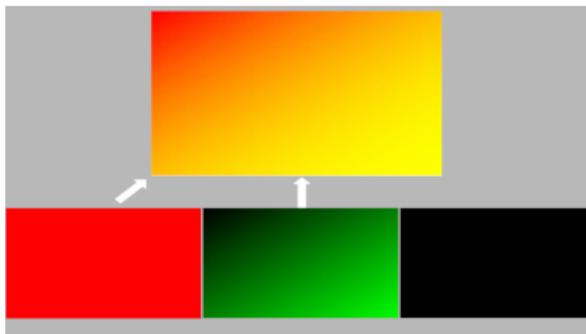
Ex.1 : dégradé du rouge au jaune (canal bleu nul)



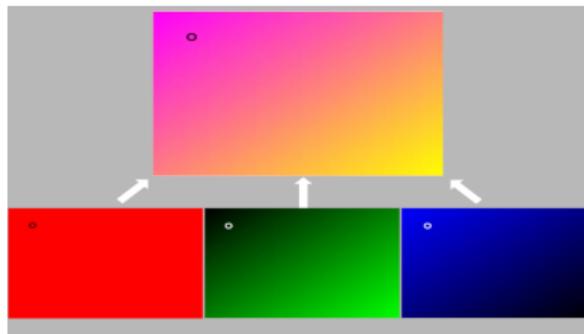
Ex.2 : idem avec dégradé du canal bleu



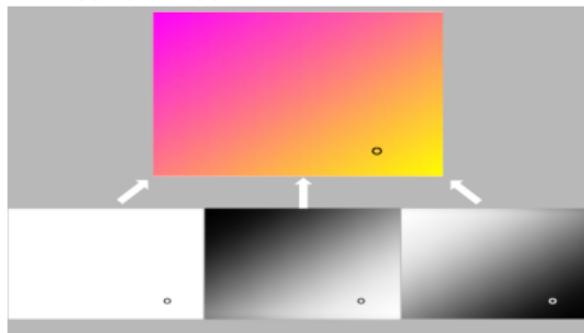
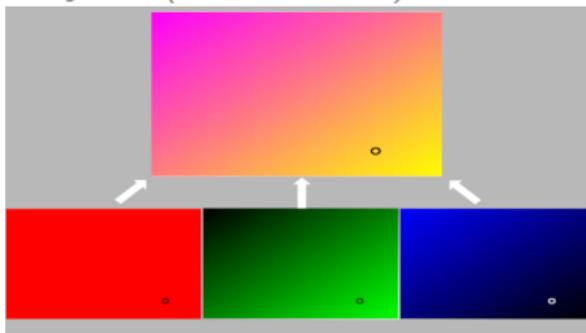
○ Canaux R, G et B d'une image couleur



Ex.1 : dégradé du rouge au jaune (canal bleu nul)



Ex.2 : idem avec dégradé du canal bleu



Représentation monochrome des canaux de l'exemple 2

★ Représentation RGB d'une image couleur

Quelle est la décomposition RGB de cette image couleur ?



★ Représentation RGB d'une image couleur

Quelle est la décomposition RGB de cette image couleur ?



★ Représentation RGB d'une image couleur

Quelle est la décomposition RGB de cette image couleur ?



A			
B			
C			

★ Représentation RGB d'une image couleur

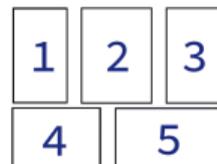
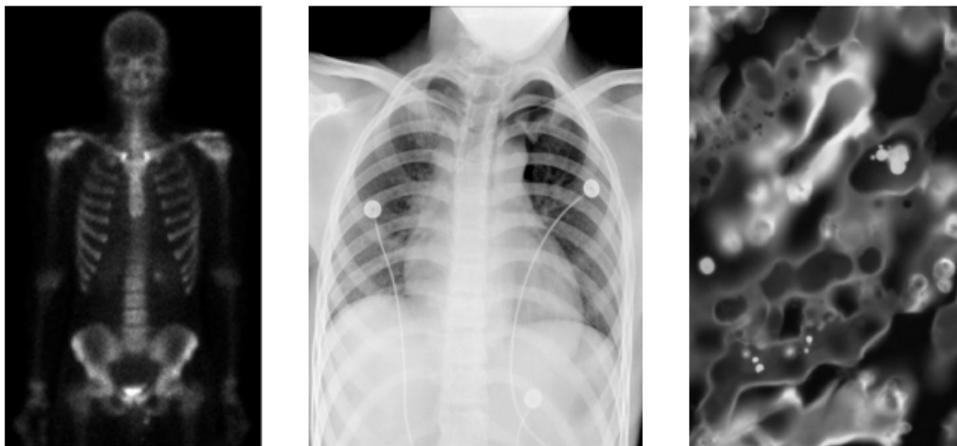
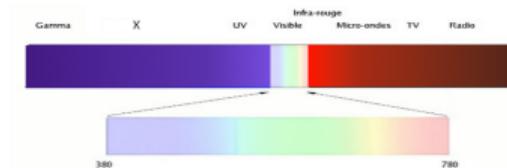
Quelle est la décomposition RGB de cette image couleur ?



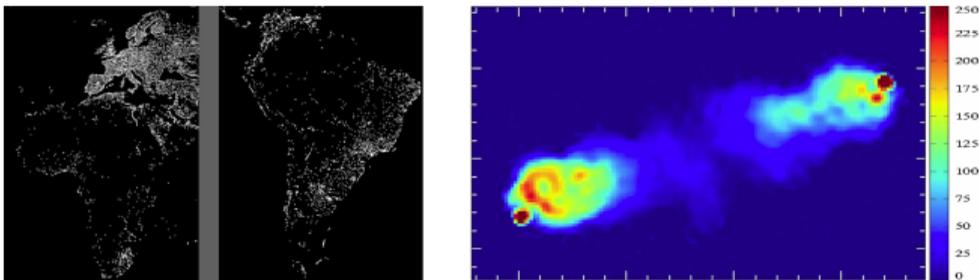
A			
B			
C			

1.7. Acquisition d'images à l'extérieur du spectre visibles

- ▶ sélection d'une bande $[\lambda_1, \lambda_2]$ du spectre électromagnétique
- ▶ DSE de λ_1 à $\lambda_2 \rightarrow I$

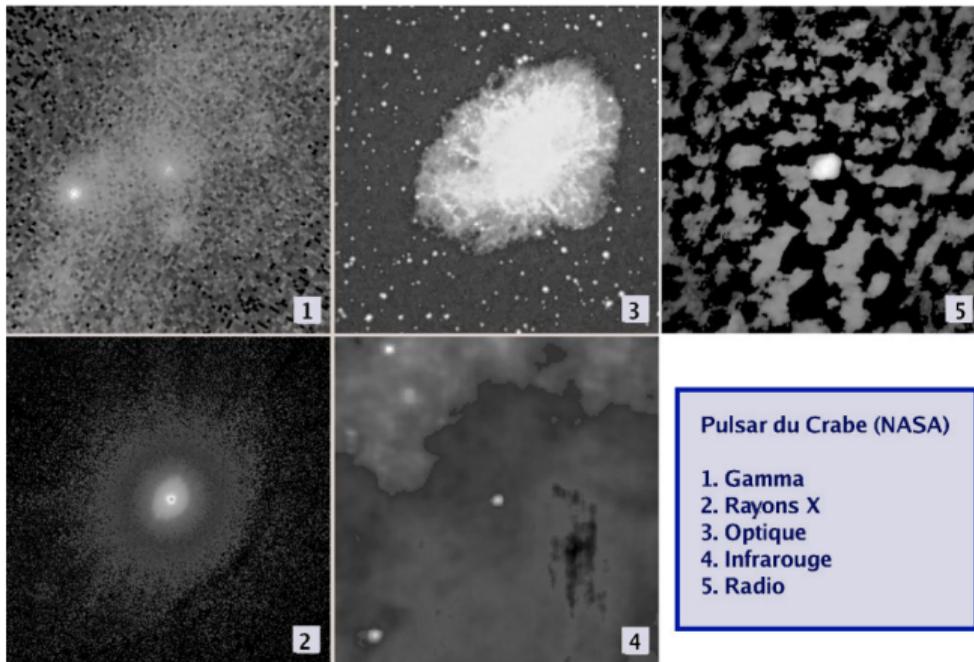


1. rayonnements gamma
2. rayonnements X
3. rayonnements UV (maïs)
4. rayonnements infrarouge
5. ondes radio (constellation du cygne)



○ Acquisition multibandes

- ▶ combinaisons d'acquisitions utilisant différentes bandes du spectre électromagnétique
- ▶ permet de capturer des phénomènes distincts
- ▶ exemple en imagerie spatiale



► imagerie satellitaire – ex : images satellites de la région de Washington, D.C. (NASA)

- (1) : 450–520nm (bleu)
- (2) : 520–600nm (vert)
- (3) : 630–690nm (rouge)

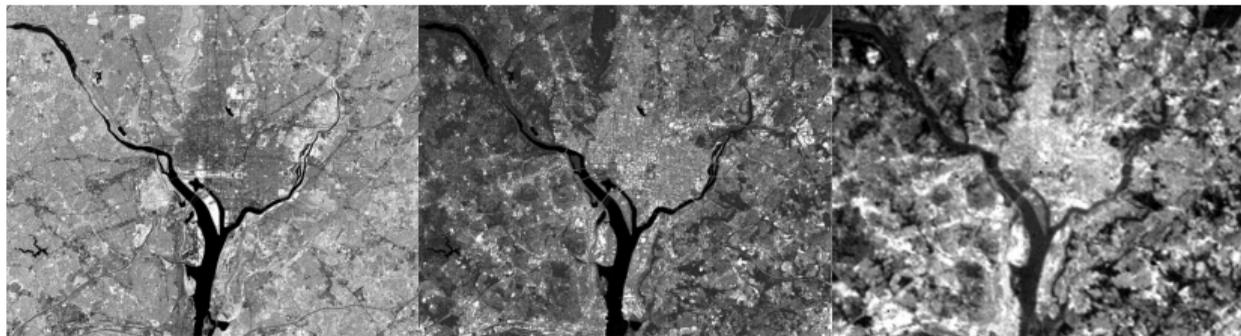


(1)

(2)

(3)

- (4) : 780–900nm (proche infrarouge)
- (6) : 2080–2350nm (infrarouge moyen)
- (7) : 10400–12500nm (infrarouge thermique)



(4)

(5)

(6)

○ Recomposition en fausses couleurs

- ▶ ex. : utilisation des acquisitions (1), (2) et (3) pour former une image couleur

(3)



→ *R*

(2)



→ *G*

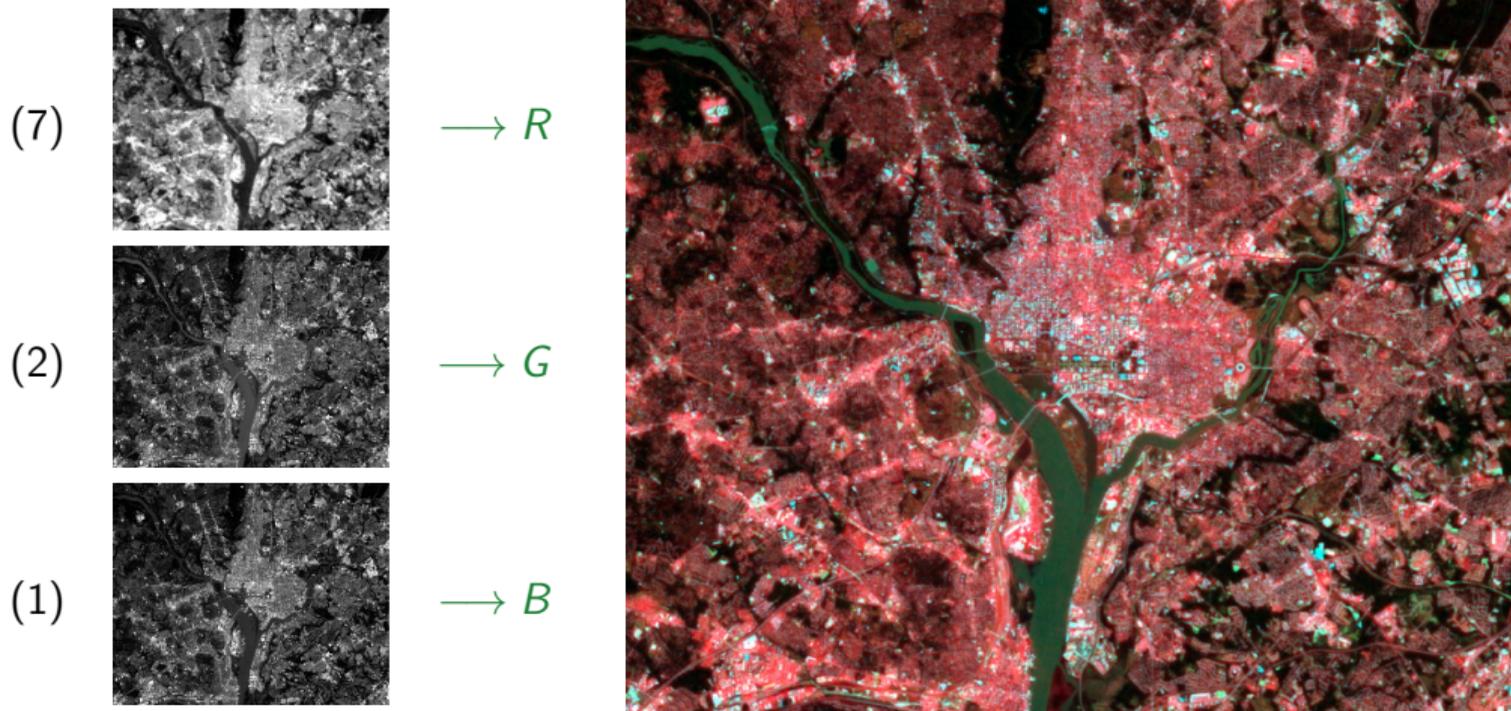
(1)



→ *B*



- ▶ alternative : utilisation de la bande infrarouge thermique (7) comme canal rouge
- ▶ recombinaison avec les bandes (2) et (1)

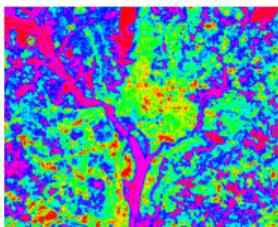


- ▶ utilisation d'autres espaces de représentation des couleurs que RGB
- ▶ infrarouge thermique (7) \rightarrow teinte (a), moyenne $\{(1), (2), (3)\}$ \rightarrow luminosité (b)

(7)



(a)

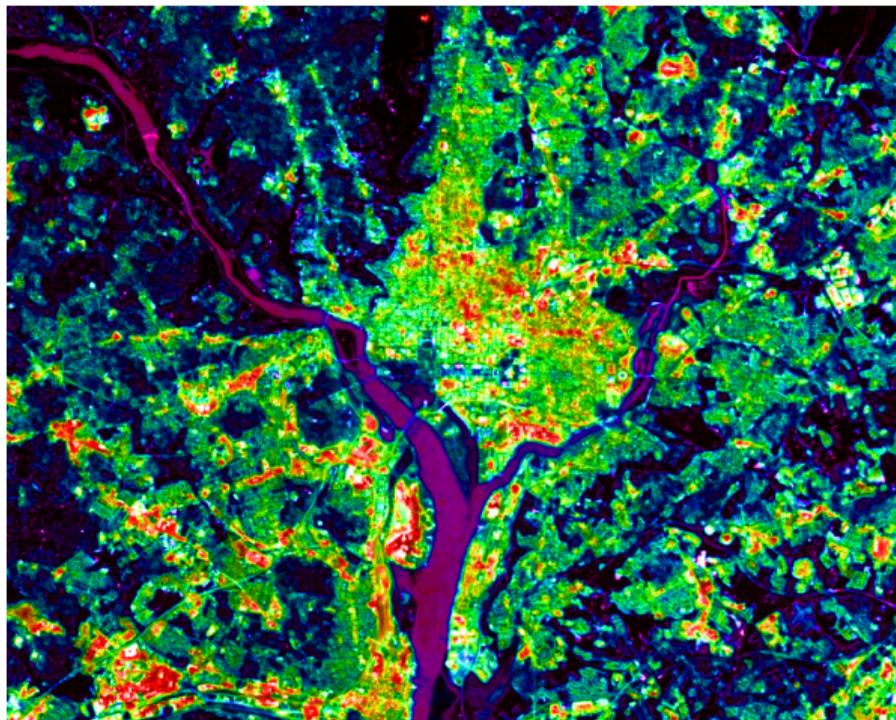


$\rightarrow H$

(b)



$\rightarrow L$



○ Autre exemple : visualisation fausses couleurs en imagerie spatiale

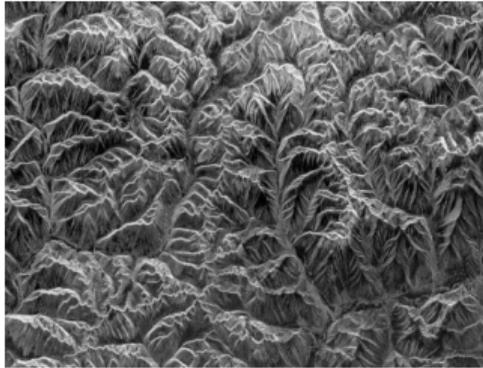
▶ supernova SN 1006

- ▶ longueurs d'onde radio : rouge
- ▶ bande visible : jaune
- ▶ rayonnements X : bleu

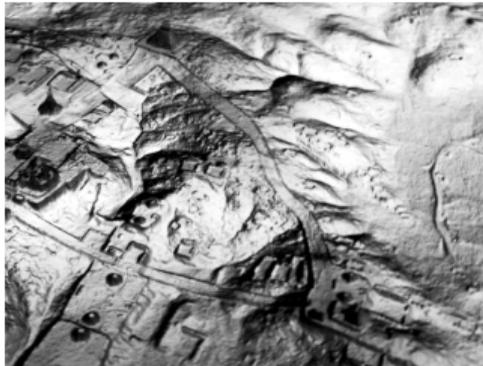


1.8. Autres dispositifs d'acquisition d'images

Mesure de distance



radar (ondes radio)

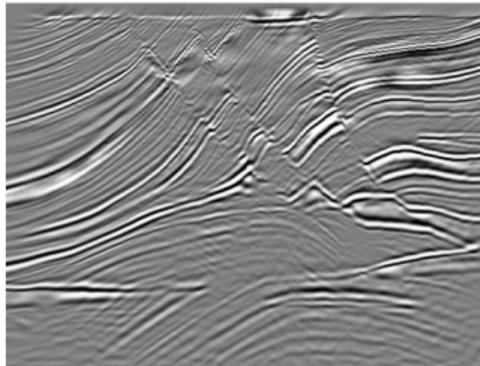


lidar (laser UV, visible et IR)

Ondes sonores

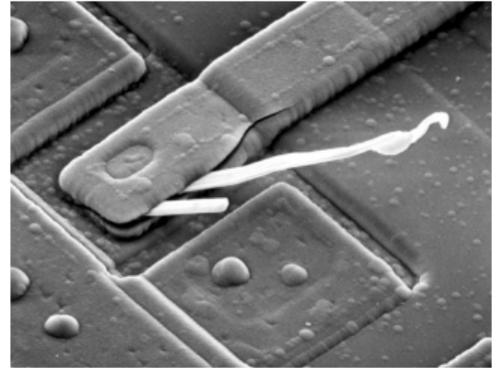


échographie



imagerie sismique

Microscopie électronique à balayage



faisceau d'électrons + capteur

Permet d'acquérir des images d'objets de taille inférieure aux longueurs d'ondes visibles.

2. Représentation spatiale des images

2.1. Représentation continue

- ▶ une image 2D continue I est un ensemble de points de \mathbb{R}^2 colorés
- ▶ l'ensemble des points est le **domaine** \mathcal{D} de l'image (généralement un rectangle)
- ▶ une image continue I est représentée par une fonction définie sur un domaine \mathcal{D} et à valeurs dans un **espace de couleurs** \mathcal{C} (voir thème A-3)

$$I : \mathcal{D} \rightarrow \mathcal{C}$$

- ▶ la représentation continue des images est utile pour **définir mathématiquement** certaines transformations (monde des modèles)

2.2. Représentation discrète (images numériques)

○ Représentation standard

- ▶ image discrète : représentation concrète utilisée la mise en œuvre de traitements
 - ▶ *domaine discret D : en général rectangle de \mathbb{Z}^2*
 - ▶ *espace de couleur discret : K canaux avec pour chacun M niveaux (valeurs possibles)*

$$I : D \rightarrow [0..M - 1]^K$$

- ▶ nombre de canaux, usuellement 1 à 4 :
 - ▶ $K = 1$: image *monochrome*
 - ▶ $K = 3$: image *couleur* (par exemple, canaux R , G et B)
 - ▶ on rajoute parfois une information de *transparence* (canal α) aux images monochrome ($K = 2$) ou couleurs ($K = 4$, $RGBA$)
 - ▶ $K > 3$ (+1) pour des acquisitions multibandes
- ▶ nombre de niveaux par canal, (usuellement puissance de 2) :
 - ▶ en acquisition : $M = 2^8, 2^{12}$ ou 2^{16} (8 à 16 bits par canal, soit resp. 256, 4096 et 65536 niveaux par canal)
 - ▶ en traitement : jusqu'à 32 bits par canal (Gimp 2.10, Photoshop, GEGL)

2.2. Représentation discrète (images numériques)

○ Représentation standard

- ▶ image discrète : représentation concrète utilisée la mise en œuvre de traitements
 - ▶ *domaine discret* D : en général rectangle de \mathbb{Z}^2
 - ▶ *espace de couleur discret* : K *canaux* avec pour chacun M *niveaux* (valeurs possibles)

$$I : D \rightarrow [0..M - 1]^K$$

- ▶ nombre de *canaux*, usuellement 1 à 4 :
 - ▶ $K = 1$: image *monochrome*
 - ▶ $K = 3$: image *couleur* (par exemple, canaux R , G et B)
 - ▶ on rajoute parfois une information de *transparence* (canal α) aux images monochrome ($K = 2$) ou couleurs ($K = 4$, $RGBA$)
 - ▶ $K > 3$ (+1) pour des acquisitions multibandes
- ▶ nombre de *niveaux* par canal, (usuellement puissance de 2) :
 - ▶ en acquisition : $M = 2^8, 2^{12}$ ou 2^{16} (8 à 16 bits par canal, soit resp. 256, 4096 et 65536 niveaux par canal)
 - ▶ en traitement : jusqu'à 32 bits par canal (Gimp 2.10, Photoshop, GEGL)

2.2. Représentation discrète (images numériques)

○ Représentation standard

- ▶ image discrète : représentation concrète utilisée la mise en œuvre de traitements
 - ▶ *domaine discret* D : en général rectangle de \mathbb{Z}^2
 - ▶ *espace de couleur discret* : K *canaux* avec pour chacun M *niveaux* (valeurs possibles)

$$I : D \rightarrow [0..M - 1]^K$$

- ▶ nombre de *canaux*, usuellement 1 à 4 :
 - ▶ $K = 1$: image *monochrome*
 - ▶ $K = 3$: image *couleur* (par exemple, canaux R , G et B)
 - ▶ on rajoute parfois une information de *transparence* (canal α) aux images monochrome ($K = 2$) ou couleurs ($K = 4$, $RGBA$)
 - ▶ $K > 3 (+1)$ pour des acquisitions multibandes
- ▶ nombre de *niveaux* par canal, (usuellement puissance de 2) :
 - ▶ en acquisition : $M = 2^8, 2^{12}$ ou 2^{16} (8 à 16 bits par canal, soit resp. 256, 4096 et 65536 niveaux par canal)
 - ▶ en traitement : jusqu'à 32 bits par canal (Gimp 2.10, Photoshop, GEGL)

○ Représentation normalisée

- ▶ valeurs des canaux codées par des réels sur l'intervalle $[0, 1]$

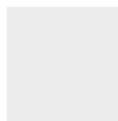
$$I' : D \rightarrow [0, 1]^K$$

- ▶ on **normalise** en divisant chaque valeur par la valeur maximale du canal

- ▶ *en monochrome* : $I'(p) = \frac{I(p)}{M-1}$

ex. pour $M = 256$:

gris moyen

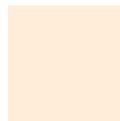


$$127 \longrightarrow \frac{127}{255} \longrightarrow 0.5$$

- ▶ *en RGB* : si $I(p) = (r, g, b)$, $I'(p) = \left(\frac{r}{M-1}, \frac{g}{M-1}, \frac{b}{M-1}\right)$

ex. pour $M = 256$:

orange



$$: (255, 127, 0) \longrightarrow (1.0, 0.5, 0.0)$$

- ▶ on **dénormalise** en multipliant par $M - 1$

○ Représentation normalisée

- ▶ valeurs des canaux codées par des réels sur l'intervalle $[0, 1]$

$$I' : D \rightarrow [0, 1]^K$$

- ▶ on **normalise** en divisant chaque valeur par la valeur maximale du canal

- ▶ *en monochrome* : $I'(p) = \frac{I(p)}{M-1}$

ex. pour $M = 256$:

gris moyen



$$127 \longrightarrow \frac{127}{255} \longrightarrow 0.5$$

- ▶ *en RGB* : si $I(p) = (r, g, b)$, $I'(p) = \left(\frac{r}{M-1}, \frac{g}{M-1}, \frac{b}{M-1}\right)$

ex. pour $M = 256$:

orange



$$: (255, 127, 0) \longrightarrow (1.0, 0.5, 0.0)$$

- ▶ on **dénormalise** en multipliant par $M - 1$

○ Représentation normalisée

- ▶ valeurs des canaux codées par des réels sur l'intervalle $[0, 1]$

$$I' : D \rightarrow [0, 1]^K$$

- ▶ on **normalise** en divisant chaque valeur par la valeur maximale du canal

- ▶ *en monochrome* : $I'(p) = \frac{I(p)}{M-1}$

ex. pour $M = 256$:

gris moyen



$$127 \longrightarrow \frac{127}{255} \longrightarrow 0.5$$

- ▶ *en RGB* : si $I(p) = (r, g, b)$, $I'(p) = \left(\frac{r}{M-1}, \frac{g}{M-1}, \frac{b}{M-1}\right)$

ex. pour $M = 256$:

orange



$$: (255, 127, 0) \longrightarrow (1.0, 0.5, 0.0)$$

- ▶ on **dénormalise** en multipliant par $M - 1$

Quels sont selon-vous les avantages de la représentation normalisée par rapport à la représentation standard ?

A : calculs plus rapide

B : calculs plus précis

C : occupation mémoire moins importante

D : plus grande puissance d'expression

Quels sont selon-vous les avantages de la représentation normalisée par rapport à la représentation standard ?

A : calculs plus rapide

B : calculs plus précis

C : occupation mémoire moins importante

D : plus grande puissance d'expression

par exemple, permet d'appliquer une fonction réelle définie dans $[0, 1]$, comme $i' = i^{\frac{1}{\gamma}}$

○ Taille et taille mémoire

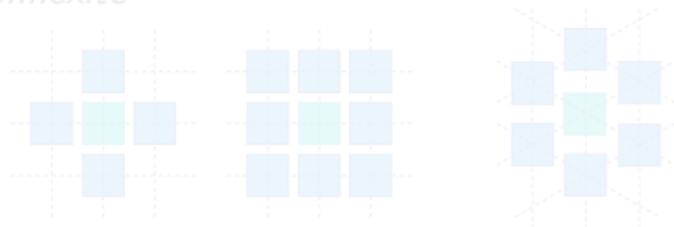
- ▶ **pixel** : point (élément du domaine D) valué (couleur, niveau de gris, etc.) —

$$\{p, I(p)\} , (p_i, p_j, p_c) , \dots$$

- ▶ **taille** de l'image : nombre de points (\neq dimension) — *en 2D : largeur \times hauteur*
- ▶ **taille mémoire** d'une image : taille \times nombre d'octets par pixel (codage de la couleur)

○ Voisinages et connexité

- ▶ **voisinage** d'un point p : $V(p)$
 - ▶ *en 2D : 4-connexité et 8-connexité*
 - ▶ *alternative : 6-connexité*



- ▶ *en 3D : 6, 18 et 26-connexité*

○ Taille et taille mémoire

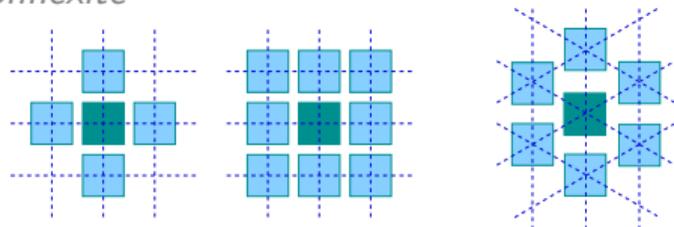
- ▶ **pixel** : point (élément du domaine D) valué (couleur, niveau de gris, etc.) —

$$\{p, I(p)\}, (p_i, p_j, p_c), \dots$$

- ▶ **taille** de l'image : nombre de points (\neq dimension) — *en 2D : largeur \times hauteur*
- ▶ **taille mémoire** d'une image : taille \times nombre d'octets par pixel (codage de la couleur)

○ Voisines et connexité

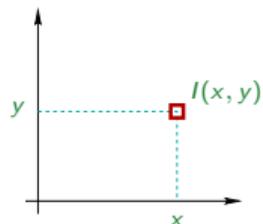
- ▶ **voisinage** d'un point p : $V(p)$
 - ▶ *en 2D : 4-connexité et 8-connexité*
 - ▶ *alternative : 6-connexité*



- ▶ *en 3D : 6, 18 et 26-connexité*

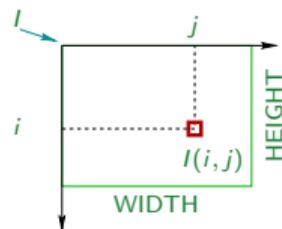
○ Accès à un point d'une image 2D

repère carthésien usuel



$I(x, y)$: notation mathématique usuelle

repère orienté vers le bas



$I(i, j)$: plus adaptée aux structures de données

- ▶ allocation mémoire du buffer d'une image

```
I = malloc(WIDTH*HEIGHT*sizeof(pixel))
```

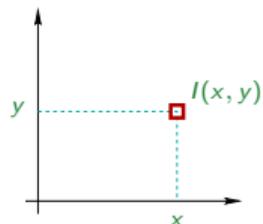
- ▶ rangement ligne par ligne : accès au pixel (i, j)

i : ligne
 j : colonne

le pixel $I[i][j]$ est à partir de l'adresse $I + i*WIDTH + j$

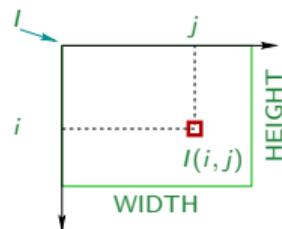
○ Accès à un point d'une image 2D

repère carthésien usuel



$I(x, y)$: notation mathématique usuelle

repère orienté vers le bas



$I(i, j)$: plus adaptée aux structures de données

- ▶ allocation mémoire du buffer d'une image

```
I = malloc(WIDTH*HEIGHT*sizeof(pixel))
```

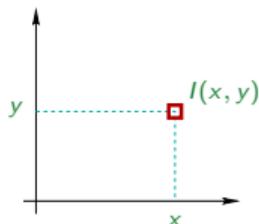
- ▶ rangement ligne par ligne : accès au pixel (i, j)

i : ligne
 j : colonne

le pixel $I[i][j]$ est à partir de l'adresse $I + i*WIDTH + j$

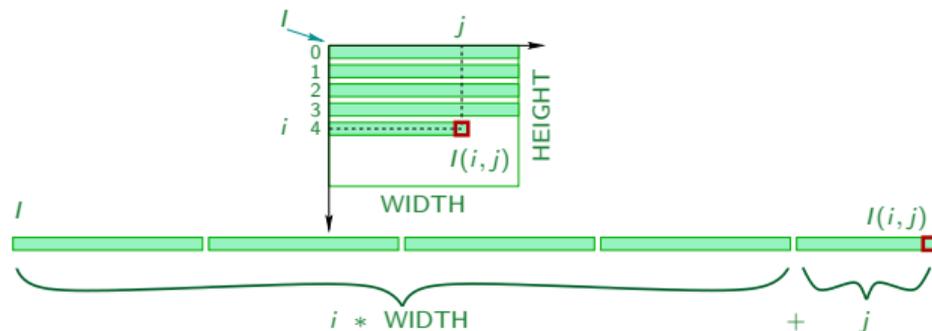
○ Accès à un point d'une image 2D

repère carthésien usuel



$I(x, y)$: notation mathématique usuelle

repère orienté vers le bas



$I(i, j)$: plus adaptée aux structures de données

- ▶ allocation mémoire du buffer d'une image

```
I = malloc(WIDTH*HEIGHT*sizeof(pixel))
```

- ▶ rangement ligne par ligne : accès au pixel (i, j)

i : ligne
 j : colonne

le pixel $I[i][j]$ est à partir de l'adresse $I + i*WIDTH + j$

★ Allocation du buffer de stockage des pixels d'une image

On implémente en C une image RGB de taille *width* par *height*, avec 8 bits par canal. Le buffer stockant le contenu de l'image est alloué dynamiquement et référencé par un pointeur de nom *image* :

```
image = malloc(buffer_size)
```

Quelle est la bonne façon d'initialiser la variable *buffer_size* ?

A	<pre>size_t buffer_size = (width*sizeof(char)) * (height*sizeof(char));</pre>
B	<pre>size_t buffer_size = width * height * sizeof(char);</pre>
C	<pre>size_t buffer_size = width * height * (3*sizeof(char));</pre>
D	<pre>size_t buffer_size = width * height * (3*sizeof(float));</pre>

★ Allocation du buffer de stockage des pixels d'une image

On implémente en C une image RGB de taille *width* par *height*, avec 8 bits par canal. Le buffer stockant le contenu de l'image est alloué dynamiquement et référencé par un pointeur de nom *image* :

```
image = malloc(buffer_size)
```

Quelle est la bonne façon d'initialiser la variable *buffer_size* ?

A	<code>size_t buffer_size = (width*sizeof(char)) * (height*sizeof(char));</code>
B	<code>size_t buffer_size = width * height * sizeof(char);</code>
C	<code>size_t buffer_size = width * height * (3*sizeof(char));</code>
D	<code>size_t buffer_size = width * height * (3*sizeof(float));</code>



Attention à bien différencier `buffer_size` et `size` (= `width*height`).

★ Parcours des pixels d'une image

On définit la fonction `offset(long width, long i, long j)` qui retourne la valeur
 $(i * \text{width} + j) * (3 * \text{sizeof}(\text{char}))$

Que fait le code suivant ?

```
for (size_t i = 0; i < height; i++)
{
    for (size_t j = 0; j < width; j++)
    {
        unsigned char *pixel = image + offset(width, i, j);

        pixel[0] = 255;
        pixel[1] = 127;
        pixel[2] = 0;
    }
}
```

Parcours de l'image		Coloration des pixels	
A	horizontalement	C	alternativement en noir, gris et blanc
B	verticalement	D	en orange

★ Parcours des pixels d'une image

On définit la fonction `offset(long width, long i, long j)` qui retourne la valeur
 $(i * \text{width} + j) * (3 * \text{sizeof}(\text{char}))$

Que fait le code suivant ?

```
for (size_t i = 0; i < height; i++)
{
    for (size_t j = 0; j < width; j++)
    {
        unsigned char *pixel = image + offset(width, i, j);

        pixel[0] = 255;
        pixel[1] = 127;
        pixel[2] = 0;
    }
}
```

Parcours de l'image		Coloration des pixels	
A	horizontalement	C	alternativement en noir, gris et blanc
B	verticalement	D	en orange

★ Optimisation du parcours des pixels d'une image

Pour parcourir l'image verticalement (en colonne), il suffit d'échanger l'ordre des deux boucles :

```
for (size_t j = 0; j < width; j++)  
{  
    for (size_t i = 0; i < height; i++)  
        ...  
}
```

Une troisième méthode consiste à balayer l'ensemble du buffer avec un pointeur :

```
for (unsigned char *pixel = image; pixel < image + buffer_size; pixel += 3)  
    ...
```

Comment estimez-vous les temps d'exécution relatifs de ces trois méthodes ?

A	$P \sim IJ \sim JI$
B	$P < IJ \sim JI$

C	$P < IJ < JI$
D	$P < JI < IJ$

IJ : temps de parcours en ligne

JI : temps parcours en colonne

P : temps balayage avec un pointeur

★ Optimisation du parcours des pixels d'une image

Pour parcourir l'image verticalement (en colonne), il suffit d'échanger l'ordre des deux boucles :

```
for (size_t j = 0; j < width; j++)  
{  
    for (size_t i = 0; i < height; i++)  
        ...  
}
```

Une troisième méthode consiste à balayer l'ensemble du buffer avec un pointeur :

```
for (unsigned char *pixel = image; pixel < image + buffer_size; pixel += 3)  
    ...
```

Comment estimez-vous les temps d'exécution relatifs de ces trois méthodes ?

A	$P \sim IJ \sim JI$
B	$P < IJ \sim JI$

C	$P < IJ < JI$
D	$P < JI < IJ$

IJ : temps de parcours en ligne

JI : temps parcours en colonne

P : temps balayage avec un pointeur

★ Optimisation du parcours des pixels d'une image

Exemple de temps d'exécution (réel, utilisateur et système) en secondes pour une image RGB de 1000000×20000 pixels (~ 18.6 GO).

	<i>P</i> (pointeur)			<i>IJ</i> (en ligne)			<i>Jl</i> (en colonne)		
	real	user	sys	real	user	sys	real	user	sys
sans optimisation de code	5.5	4.3	1.2	8.6	7.1	1.3	20.0	18.8	1.7
optimisation : <code>gcc -O4</code>	1.5	0.2	1.3	2.9	1.6	1.3	18.6	16.5	1.8

Conditions de test :

- ▶ mesures effectuées sous GNU/Linux avec la commande `time`
- ▶ configuration :
 - Processeur : Intel(R) Core(TM) i5-10400F
 - Mémoire : 32 GO (dont 5.6 GO attribués au processus)
 - Système : Fedora 33 5.9.16-200.fc33.x86_64
 - Compilateur : gcc 10.2.1

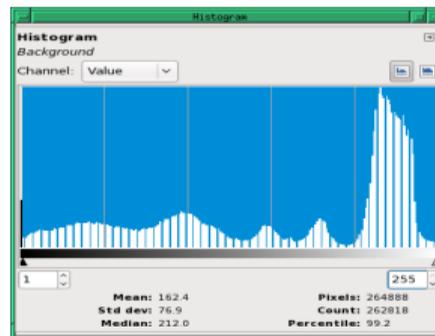
3. Calcul de paramètres statistiques

- ▶ modélisation de structures géométriques (bruit, texture)
- ▶ modélisation de caractéristiques visuelles (clarté, contraste)
 - ⇒ paramétrage d'algorithmes

3.1. Histogramme d'un canal

$$h : [0 .. M-1] \rightarrow \mathbb{N}$$

$h(i)$ est le nombre de pixels de valeur i



3.2. Histogramme normalisé et histogramme cumulé

- ▶ l'histogramme d'un canal $h(i)$ indique le nombre de pixels de niveau i
- ▶ la somme des valeurs de $h(i)$ donne le nombre N de pixels (la taille) de l'image :

$$N = \sum_{i=0}^{M-1} h(i)$$

- ▶ l'histogramme normalisé $h'(i)$ exprime le pourcentage de pixels de chaque niveau (à un facteur 100)

$$h'(i) = \frac{h(i)}{\sum_{i=0}^{M-1} h(i)}$$

- ▶ l'histogramme cumulé $H(i)$ indique le nombre de pixels de niveau $\leq i$

$$H(i) = \sum_{j=0}^i h(j)$$

- ▶ on utilise parfois aussi l'histogramme normalisé cumulé $H'(i)$

3.2. Histogramme normalisé et histogramme cumulé

- ▶ l'histogramme d'un canal $h(i)$ indique le **nombre** de pixels de niveau i
- ▶ la somme des valeurs de $h(i)$ donne le nombre N de pixels (la taille) de l'image :

$$N = \sum_{i=0}^{M-1} h(i)$$

- ▶ l'**histogramme normalisé** $h'(i)$ exprime le **pourcentage** de pixels de chaque niveau (à un facteur 100)

$$h'(i) = \frac{h(i)}{\sum_{i=0}^{M-1} h(i)}$$

- ▶ l'**histogramme cumulé** $H(i)$ indique le nombre de pixels de niveau $\leq i$

$$H(i) = \sum_{j=0}^i h(j)$$

- ▶ on utilise parfois aussi l'**histogramme normalisé cumulé** $H'(i)$

3.2. Histogramme normalisé et histogramme cumulé

- ▶ l'histogramme d'un canal $h(i)$ indique le **nombre** de pixels de niveau i
- ▶ la somme des valeurs de $h(i)$ donne le nombre N de pixels (la taille) de l'image :

$$N = \sum_{i=0}^{M-1} h(i)$$

- ▶ l'**histogramme normalisé** $h'(i)$ exprime le **pourcentage** de pixels de chaque niveau (à un facteur 100)

$$h'(i) = \frac{h(i)}{\sum_{i=0}^{M-1} h(i)}$$

- ▶ l'**histogramme cumulé** $H(i)$ indique le nombre de pixels de niveau $\leq i$

$$H(i) = \sum_{j=0}^i h(j)$$

- ▶ on utilise parfois aussi l'**histogramme normalisé cumulé** $H'(i)$

3.3. Interprétation probabiliste de l'histogramme

- ▶ la somme des valeurs de $h'(i)$ vaut toujours 1

$$\begin{aligned}\sum_{i=0}^{M-1} h'(i) &= \sum_{i=0}^{M-1} \frac{h(i)}{\sum_{i=0}^{M-1} h(i)} \\ &= \frac{\sum_{i=0}^{M-1} h(i)}{\sum_{i=0}^{M-1} h(i)} \\ &= 1\end{aligned}$$

- ▶ $h'(i)$ peut être vue comme une **loi de probabilité** : c'est la probabilité de choisir au hasard un pixel de niveau i dans l'image
- ▶ permet d'associer des **paramètres statistiques** à une image : moyenne, variance, etc.

3.3. Interprétation probabiliste de l'histogramme

- ▶ la somme des valeurs de $h'(i)$ vaut toujours 1

$$\begin{aligned}\sum_{i=0}^{M-1} h'(i) &= \sum_{i=0}^{M-1} \frac{h(i)}{\sum_{i=0}^{M-1} h(i)} \\ &= \frac{\sum_{i=0}^{M-1} h(i)}{\sum_{i=0}^{M-1} h(i)} \\ &= 1\end{aligned}$$

- ▶ $h'(i)$ peut être vue comme une **loi de probabilité** : c'est la probabilité de choisir au hasard un pixel de niveau i dans l'image
- ▶ permet d'associer des **paramètres statistiques** à une image : moyenne, variance, etc.

★ Moyenne et écart type d'une image

1



2



3



Les trois images sont codées sur 256 niveaux de gris

- ▶ deux images ont une moyenne μ de 128 et la troisième de 192
- ▶ deux images ont un écart σ de 92 et la troisième de 16

Retrouver les paramètres statistiques de chaque image.

	1		2		3	
	μ	σ	μ	σ	μ	σ
A	192	92	128	16	128	92
B	128	16	128	92	192	92
C	128	16	192	92	128	92
D	192	16	128	92	128	92

★ Moyenne et écart type d'une image

1



2



3



Les trois images sont codées sur 256 niveaux de gris

- ▶ deux images ont une moyenne μ de 128 et la troisième de 192
- ▶ deux images ont un écart σ de 92 et la troisième de 16

Retrouver les paramètres statistiques de chaque image.

	1		2		3	
	μ	σ	μ	σ	μ	σ
A	192	92	128	16	128	92
B	128	16	128	92	192	92
C	128	16	192	92	128	92
D	192	16	128	92	128	92

3.4. Paramètres statistiques de l'image

- ▶ **moyenne** (ou espérance) : assimilable à la **luminosité moyenne** (image monochrome ou canal de luminosité)

$$\mu = \sum_{i=0}^{M-1} h'(i) \times i \quad \left(= \frac{\sum_{i=0}^{M-1} h(i) * i}{N} \right)$$

- ▶ **variance et écart type** : assimilable au **contraste**

$$V = \sum_{i=0}^{M-1} h'(i) \times (i - \mu)^2 \quad \text{et} \quad \sigma = \sqrt{V}$$

- ▶ **entropie** : quantité d'information

$$- \sum_{i=0}^{M-1} h'(i) \times \log_2(h'(i)) \quad (\text{convention : } 0 \log_2(0) = 0)$$

- ▶ quantités calculables **localement** autour de chaque pixel de l'image : on calcule l'histogramme $h_p(i)$ sur un voisinage de chaque pixel p

3.4. Paramètres statistiques de l'image

- ▶ **moyenne** (ou espérance) : assimilable à la **luminosité moyenne** (image monochrome ou canal de luminosité)

$$\mu = \sum_{i=0}^{M-1} h'(i) \times i \quad \left(= \frac{\sum_{i=0}^{M-1} h(i) * i}{N} \right)$$

- ▶ **variance** et **écart type** : assimilable au **contraste**

$$V = \sum_{i=0}^{M-1} h'(i) \times (i - \mu)^2 \quad \text{et} \quad \sigma = \sqrt{V}$$

- ▶ **entropie** : quantité d'information

$$- \sum_{i=0}^{M-1} h'(i) \times \log_2(h'(i)) \quad (\text{convention : } 0 \log_2(0) = 0)$$

- ▶ quantités calculables **localement** autour de chaque pixel de l'image : on calcule l'histogramme $h_p(i)$ sur un voisinage de chaque pixel p

3.4. Paramètres statistiques de l'image

- ▶ **moyenne** (ou espérance) : assimilable à la **luminosité moyenne** (image monochrome ou canal de luminosité)

$$\mu = \sum_{i=0}^{M-1} h'(i) \times i \quad \left(= \frac{\sum_{i=0}^{M-1} h(i) * i}{N} \right)$$

- ▶ **variance** et **écart type** : assimilable au **contraste**

$$V = \sum_{i=0}^{M-1} h'(i) \times (i - \mu)^2 \quad \text{et} \quad \sigma = \sqrt{V}$$

- ▶ **entropie** : quantité d'information

$$- \sum_{i=0}^{M-1} h'(i) \times \log_2(h'(i)) \quad (\text{convention : } 0 \log_2(0) = 0)$$

- ▶ quantités calculables **localement** autour de chaque pixel de l'image : on calcule l'histogramme $h_p(i)$ sur un voisinage de chaque pixel p

3.4. Paramètres statistiques de l'image

- ▶ **moyenne** (ou espérance) : assimilable à la **luminosité moyenne** (image monochrome ou canal de luminosité)

$$\mu = \sum_{i=0}^{M-1} h'(i) \times i \quad \left(= \frac{\sum_{i=0}^{M-1} h(i) * i}{N} \right)$$

- ▶ **variance** et **écart type** : assimilable au **contraste**

$$V = \sum_{i=0}^{M-1} h'(i) \times (i - \mu)^2 \quad \text{et} \quad \sigma = \sqrt{V}$$

- ▶ **entropie** : quantité d'information

$$- \sum_{i=0}^{M-1} h'(i) \times \log_2(h'(i)) \quad (\text{convention : } 0 \log_2(0) = 0)$$

- ▶ quantités calculables **localement** autour de chaque pixel de l'image : on calcule l'histogramme $h_p(i)$ sur un voisinage de chaque pixel p

○ Exemple



Moyennes



Écarts types



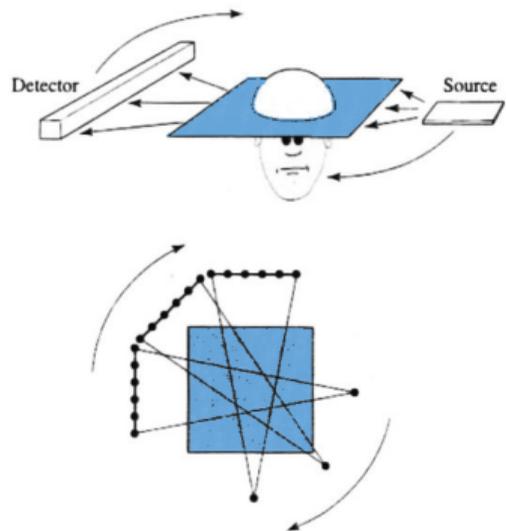
Entropies



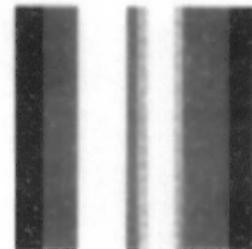
ANNEXE A

Imagerie volumique

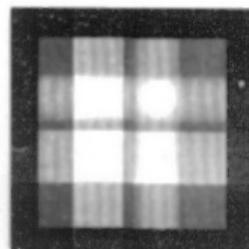
○ Tomographie rayon X



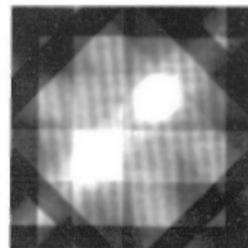
Plan à reconstruire



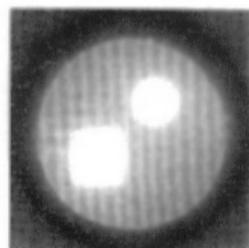
1 projection



2 projections



8 projections



32 projections

- ▶ acquisitions par **coupes 2D** parallèles
- ▶ le détecteur mesure la **somme des rayonnements absorbée** depuis la source
- ▶ exemple sur un « fantôme » avec trois niveaux d'absorption (faible, moyen, élevé)
- ▶ chaque nouvelle mesure précise la connaissance des densités sur un pla

○ Imagerie par résonance magnétique nucléaire (IRM)

- ▶ acquisition par coupes
- ▶ technique non invasive
- ▶ forte résolution en contraste (différenciation des tissus)
- ▶ champ magnétique modifiant les propriétés quantiques des noyaux atomiques : alignement des moments magnétiques (spin)
- ▶ superposition d'un second champ oscillant : émission d'un signal électromagnétique
- ▶ localisation précise de chaque source d'émission

