



DIRO
IFT 6150

TRAITEMENT D'IMAGES TEXTURES ET TRAITEMENT D'IMAGES COULEURS

Max Mignotte

Département d'Informatique et de Recherche Opérationnelle.
Http : [//www.iro.umontreal.ca/~mignotte/ift6150](http://www.iro.umontreal.ca/~mignotte/ift6150)
E-mail : mignotte@iro.umontreal.ca

TEXTURES ET TRAITEMENT D'IMAGES COULEURS SOMMAIRE

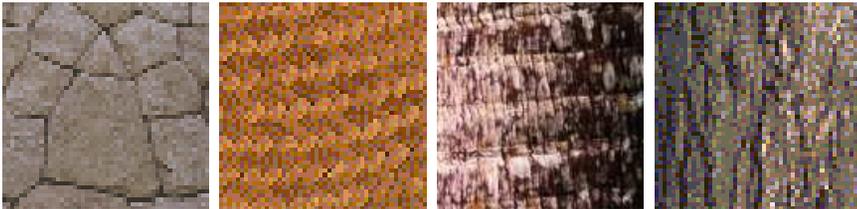
Textures

Exemples	2
Caractérisation de Texture	3

Traitement d'Images Couleurs

Représentation de la Couleur	7
Expérience de la CIE	9
Système de Référence de la CIE	10
Diagramme de Chromacité	11
Cube RGB	13
Espace des Couleurs CMY(K)	14
Espace des Couleurs HLS - HSV	16
Image RGB ► NG	17
Détection de Contours	18
Égalisation d'histogramme	19
Pseudo-Couleurs	20
Compression des Couleurs	21

TEXTURES EXEMPLES



Exemples de Textures

TEXTURES CARACTÉRISATION DE TEXTURE



(a)



► Détection, Segmentation, Recherche et Reconnaissance de texture, Indexation d'Images, etc.

Statistiques du Premier Ordre

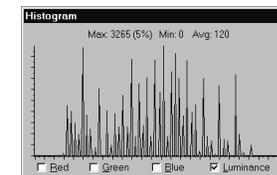
$$\mu_1 = \sum_{k=0}^{255} k p_k \quad \text{Moyenne}$$

$$\mu_2 = \sigma^2 = \sum_{k=0}^{255} (k - \mu_1)^2 p_k \quad \text{Variance}$$

$$\mu_3 = \frac{1}{\sigma^3} \sum_{k=0}^{255} (k - \mu_1)^3 p_k \quad \text{Moment d'ordre 3}$$

$$\mu_4 = \frac{1}{\sigma^4} \sum_{k=0}^{255} (k - \mu_1)^4 p_k - 3 \quad \text{Moment d'ordre 4 (Kurtosis)}$$

p_k : Probabilité d'avoir un niveau de gris égale à k (cf. Histogramme)



TEXTURES CARACTÉRISATION DE TEXTURE

Longueurs des plages de luminance constante

0	0	1	2
2	1	1	1
0	2	3	3
3	0	3	3

k	1	2	3	4
0	2	1	0	0
1	1	0	1	0
2	3	0	0	0
3	1	2	0	0

$\theta = 0^\circ$

$$\text{Longueur Moyenne} = \frac{1}{T} \sum_k \sum_l k q(k, l)$$

$$\text{Richesse en plages longues} = \frac{1}{T} \sum_k \sum_l k^2 q(k, l)$$

$$\text{Richesse en plages courtes} = \frac{1}{T} \sum_k \sum_l \frac{1}{k^2} q(k, l)$$

$$\text{Non-uniformité de luminance} = \frac{1}{T} \sum_k \left[\sum_l q(k, l) \right]^2$$

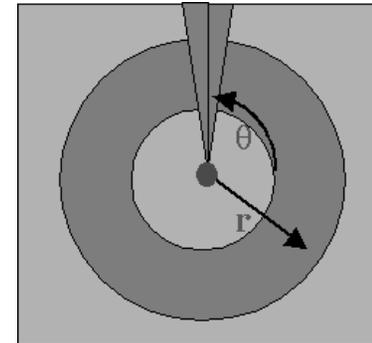
$$\text{Non-uniformité de longueur} = \frac{1}{T} \sum_l \left[\sum_k q(k, l) \right]^2$$

$$\text{Taux de plage} = \frac{1}{MN} \sum_k \sum_l q(k, l)$$

Avec $T = \sum_k \sum_l q(k, l)$

TEXTURES CARACTÉRISATION DE TEXTURE

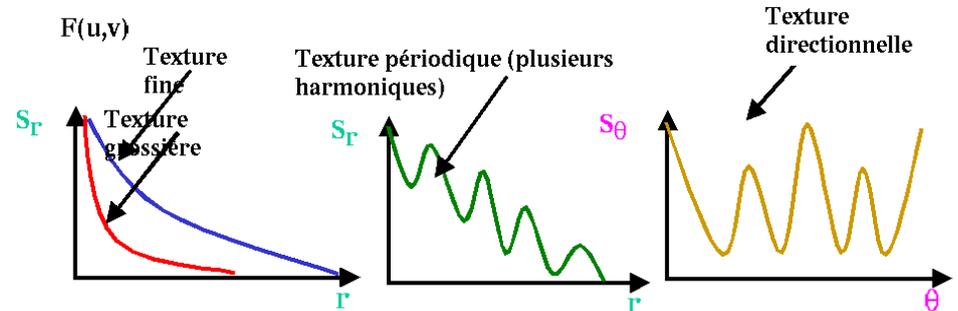
Caractéristiques Spectrales



$F(u, v)$

$$S_r(r) = \int_0^\pi |F(r, \theta)| d\theta$$

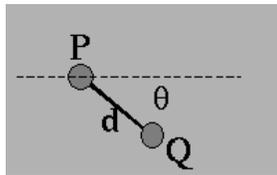
$$S_\theta(\theta) = \int_0^R |F(r, \theta)| dr$$



TEXTURES

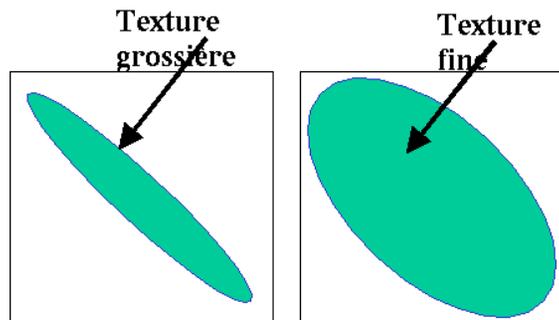
CARACTÉRISATION DE TEXTURE

Statistiques du second ordre
(Matrice de co-occurrence)



$$M(d, \theta) = [P_{i,j}]$$

$P_{i,j}$ = Probabilité[Luminance $P = i$, luminance $Q = j$]



$$\text{Énergie} = \sum_i \sum_j P_{i,j}^2$$

$$\text{Entropie} = H = - \sum_i \sum_j P_{i,j} \log P_{i,j}$$

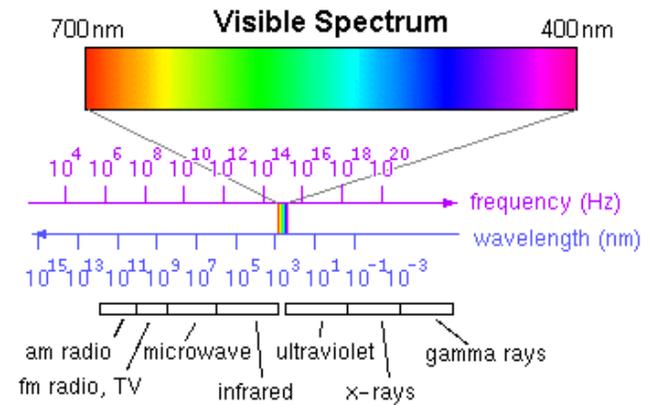
$$\text{Probabilité max} = \max[P_{i,j}]$$

$$\text{Contraste} = C = \sum_i \sum_j |i - j|^2 P_{i,j}$$

6

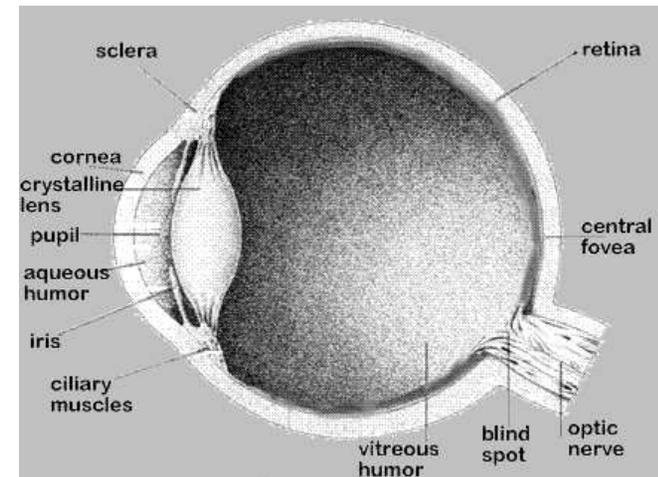
TRAITEMENT D'IMAGES COULEURS

REPRÉSENTATION DE LA COULEUR



la couleur correspond à une énergie électromagnétique comprise entre 380 et 780 nm

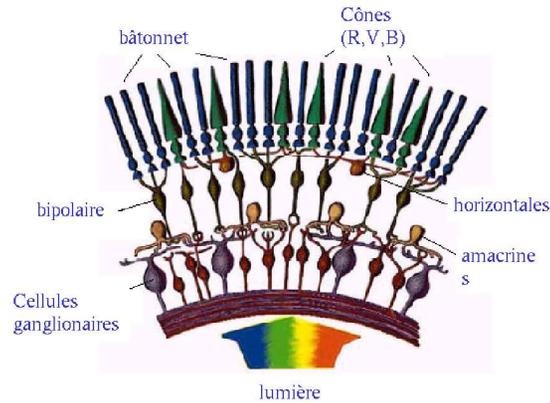
L'oeil



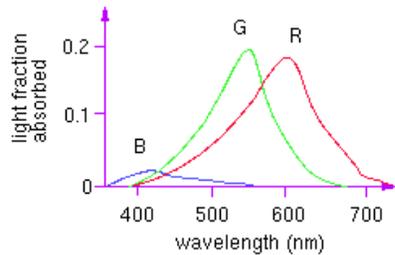
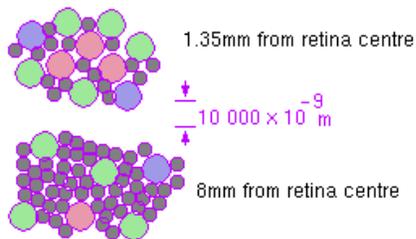
7

TRAITEMENT D'IMAGES COULEURS REPRÉSENTATION DE LA COULEUR

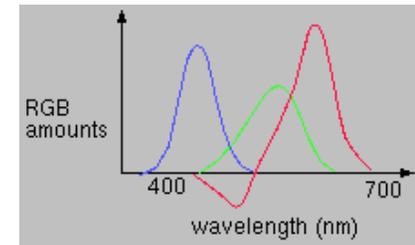
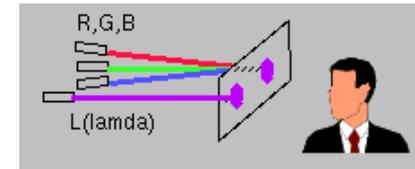
La rétine



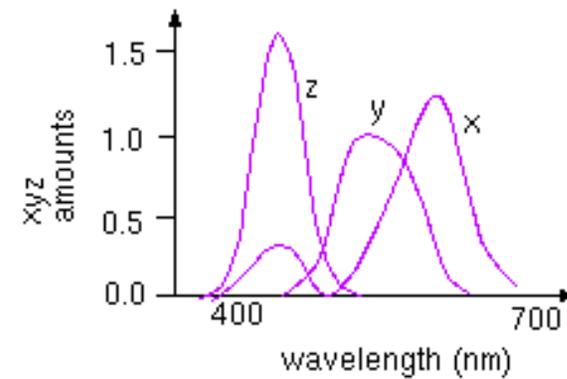
- Seulement les cônes sont responsables de la perception de la couleur



TRAITEMENT D'IMAGES COULEURS EXPÉRIENCE DE LA CIE

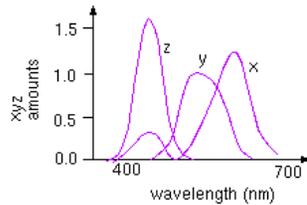


- On peut obtenir pratiquement toutes les couleurs à partir de 3 couleurs primaires monochromatiques [435.8 (bleu), 546.1 (vert), et 700 (rouge) nm] MAIS réponse négative possible pour le rouge !



- Afin d'avoir une représentation utilisant des coefficients de mélange positif, la CIE a définie trois source hypothétiques x, y, z (couleurs primaires virtuelles) ayant les spectres $R(\lambda), V(\lambda), B(\lambda)$.

TRAITEMENT D'IMAGES COULEURS SYSTÈME DE RÉFÉRENCE DE LA CIE



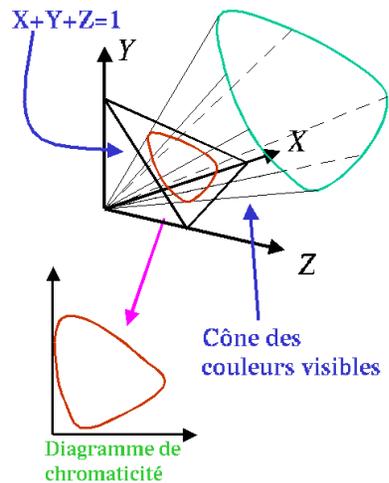
► Soit un spectre $E(\lambda)$,
les coefficients de mélange X, Y, Z sont données par

$$X = \int R(\lambda)E(\lambda) d\lambda$$

$$Y = \int V(\lambda)E(\lambda) d\lambda$$

$$Z = \int B(\lambda)E(\lambda) d\lambda$$

► Les poids X, Y, Z forment l'espace 3D de la CIE



TRAITEMENT D'IMAGES COULEURS DIAGRAMME DE CHROMATICITÉ

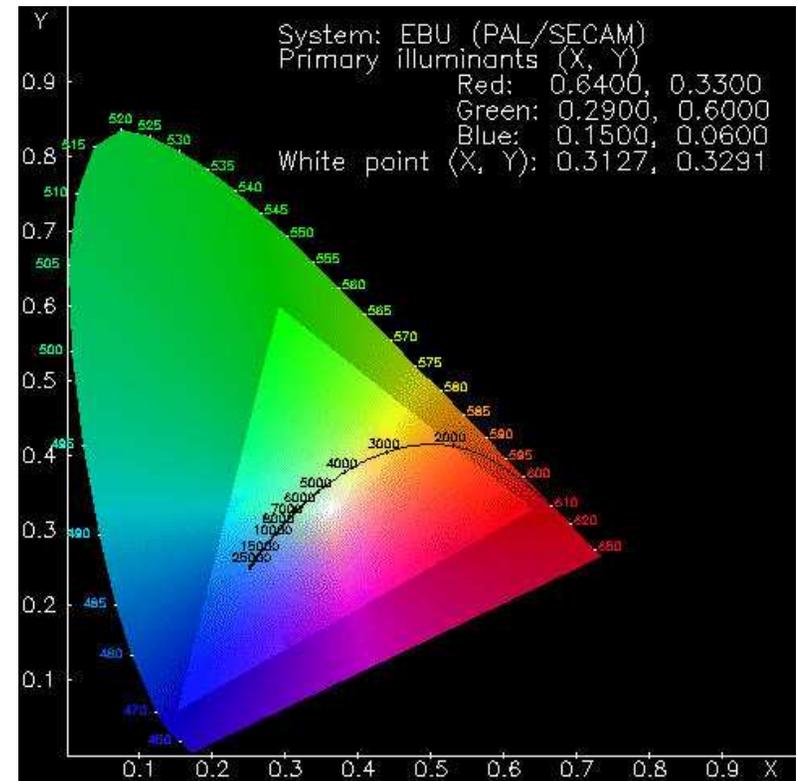
► Plus efficace de travailler
dans un espace des couleurs 2D (x, y)

$$\text{Luminance} = X + Y + Z$$

$$x = X/(X + Y + Z)$$

$$y = Y/(X + Y + Z)$$

$$z = Z/(X + Y + Z) = 1 - x - y$$



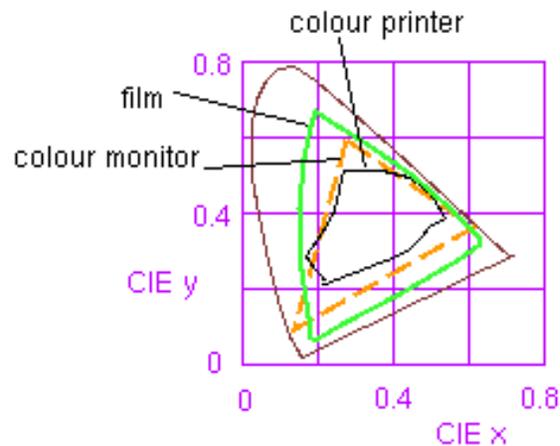
TRAITEMENT D'IMAGES COULEURS DIAGRAMME DE CHROMACITÉ

Exemples

Source	x	y
Lampe fluorescente ..	0.35	0.37
Soleil	0.32	0.33
Phosphore rouge	0.68	0.32

Gamut de Couleur

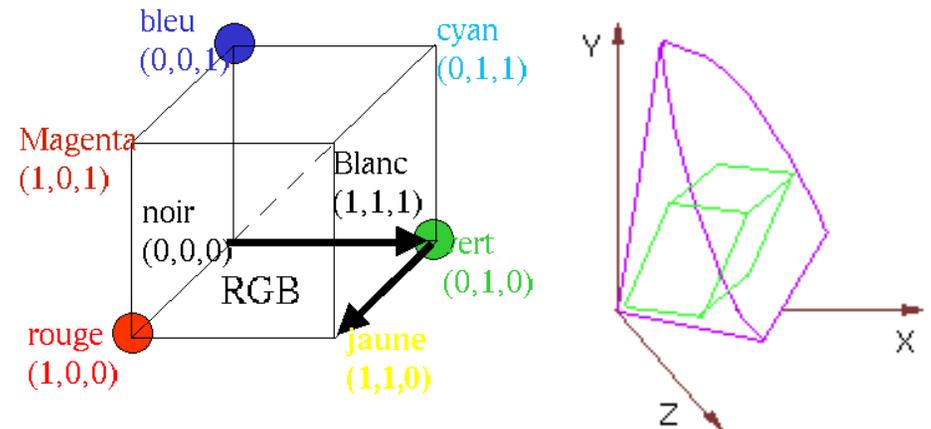
► Le diagramme de chromacité peut être utilisé pour comparer le "Gamut" de différents médiums (i.e., imprimante, moniteurs, etc.)



Remarque

Une imprimante ne peut reproduire toutes les couleurs visibles d'un moniteur

TRAITEMENT D'IMAGES COULEURS CUBE RGB



Le modèle de couleur additive est représenté par le cube RGB. Ce cube est compris dans l'espace des couleurs de la CIE.

Affichage de Couleur

Un papier vert est vert car celui-ci reflète le vert et absorbe les autres couleurs.

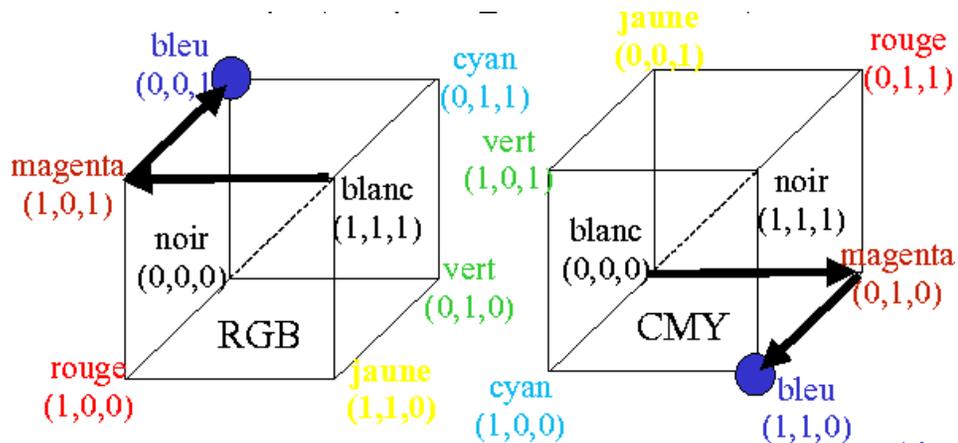
Couleur	absorbe	réfléchie
cyan	rouge	bleu et vert
magenta	vert	bleu et rouge
jaune	bleu	rouge et vert
noir	tous	aucun



TRAITEMENT D'IMAGES COULEURS ESPACE DE COULEUR CMY(K) (1)

► Utilisé par les imprimantes

Espace CMY



$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \left(\underbrace{\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}}_{\text{bleu}_{CMY}} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{\text{bleu}_{RGB}} \right)$$

► Dans l'espace RGB, on commence avec le blanc et on soustrait la qté voulue de vert (avec ajout d'encre magenta) puis de rouge (avec encre cyan, etc.)

TRAITEMENT D'IMAGES COULEURS ESPACE DE COULEUR CMY(K) (2)

Espace $C_k M_k Y_k K$

K désigne l'encre noire

1- Transformation RGB ► CMY

2 – $K = \min(C, M, Y)$

3 – $\begin{bmatrix} C_k \\ M_k \\ Y_k \end{bmatrix} = \begin{bmatrix} C \\ M \\ Y \end{bmatrix} - \begin{bmatrix} K \\ K \\ K \end{bmatrix}$

Exemple

Réalisation un gris légèrement jaune, i.e., $\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \underbrace{\begin{bmatrix} 0.5 \\ 0.5 \\ 0.6 \end{bmatrix}}_{\text{CMY}}$

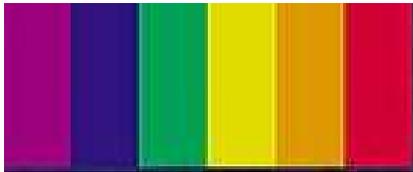
$$K = \min(C, M, Y) = \min(0.5, 0.5, 0.6) = 0.5$$

$$\begin{bmatrix} C_k \\ M_k \\ Y_k \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.6 \end{bmatrix} - \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0.1 \end{bmatrix} \leftrightarrow \begin{bmatrix} C_k \\ M_k \\ Y_k \\ K \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0.1 \\ 0.5 \end{bmatrix}$$

Si on veut générer un gris moyen légèrement jaune, on utilise la cartouche d'encre noire pour la composante grise (0.5) et un peu de jaune (0.1) au lieu d'utiliser les trois cartouches C, M, Y

TRAITEMENT D'IMAGES COULEURS

ESPACE DE COULEUR HLS HSV



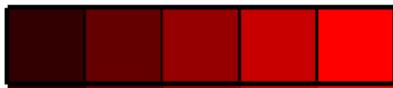
Teinte (*Hue*)

- Déterminé par la longueur d'onde dominante
- Rouge, jaune, Pourpre



Saturation

- Distance du gris de même intensité
- De gris (blanc) à pure



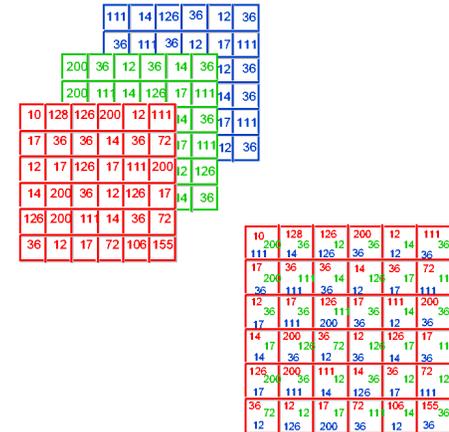
Luminance (*Lightness*)

- Intensité de lumière que la couleur réfléchit/transmet
- Brillance (*brightness*) pour émission

► Compréhension des différentes couleurs plus intuitive

TRAITEMENT D'IMAGES COULEURS

IMAGE RGB ► NG



RGB



G

(R+G+B) / 3



TRAITEMENT D'IMAGES COULEURS DETECTION DE CONTOURS



Original

R - Edges



G - Edges



B - Edges



All - Edges



TRAITEMENT D'IMAGES COULEURS ÉGALISATION D'HISTOGRAMME



Image originale



Sur R, V, B individuellement



À partir de $I = (R + V + B)/3$

TRAITEMENT D'IMAGES COULEURS PSEUDO-COULEURS

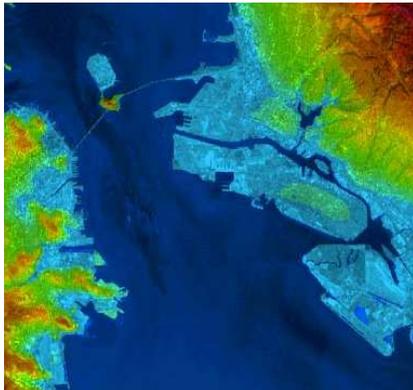
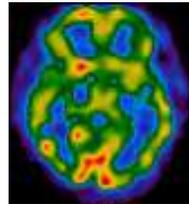
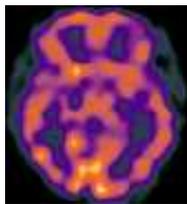
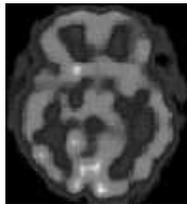
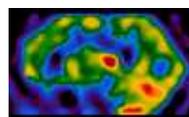
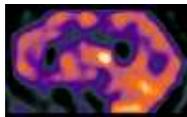


Image SPOT de la baie de San Francisco avec couleur = altitude



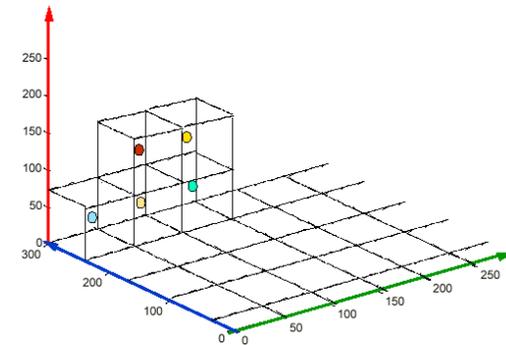
Transaxial



Coronal

TRAITEMENT D'IMAGES COULEURS COMPRESSION DES COULEURS (1)

Quantification indépendante de l'image



Original



125 Colors



64 Colors



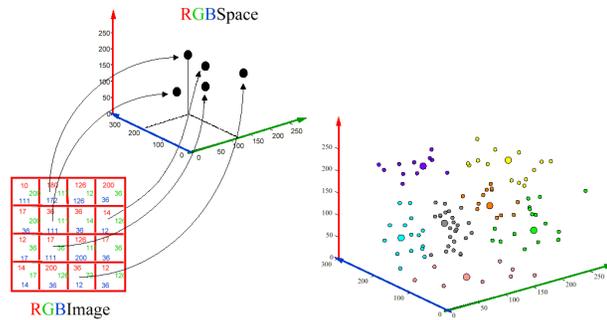
27 Colors



TRAITEMENT D'IMAGES COULEURS

COMPRESSION DES COULEURS (2)

Quantification dépendante de l'image



Principe On divise l'espace RGB en fonction de la distribution des pixels dans cette espace

- On élague les bords du cube RGB qui sont vides
- On partage l'espace restant en deux parties égales en nombre de points (partage médian)
- On répète les deux étapes récursivement
- On obtient 4, 8, ..., boîtes qui contiennent chacune à peu près le même nombre de pixels et dont la taille est inversement proportionnelle à leur densité
- On affiche pour tous les pixels de la boîte la couleur associée au centre de la boîte ou la moyenne des couleurs des pixels qu'elle contient

