Indexation d'images par contenu (1)

Content Based Image Retrieval (CBIR)

A. ELHASSOUNY

GL
ENSIAS

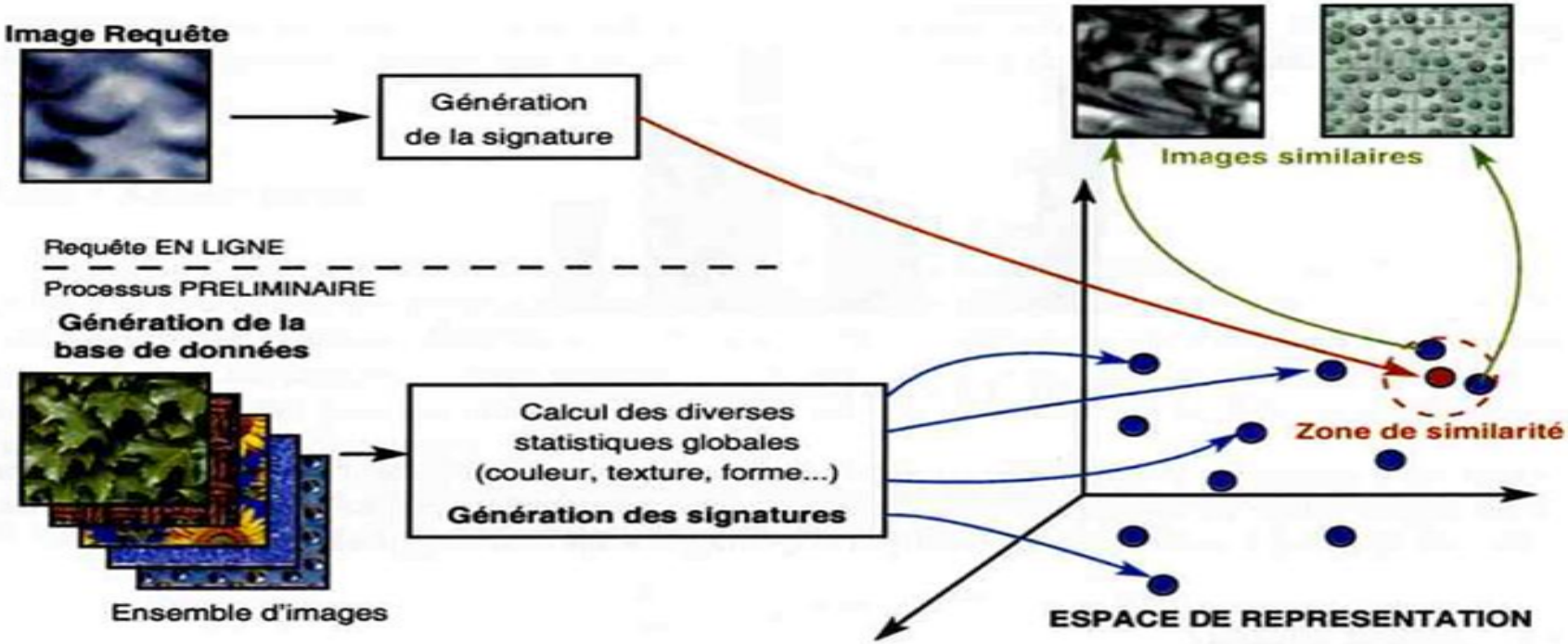
Master B2dS

PLAN

1. Indexation d'image par contenu (CBIR)
   - Principe de CBIR
   - Qu'est-ce qu'une image ?
2. Descripteurs d'image
   - Types de descripteurs
   - Descripteurs globaux

CBIR

Descripteurs globaux

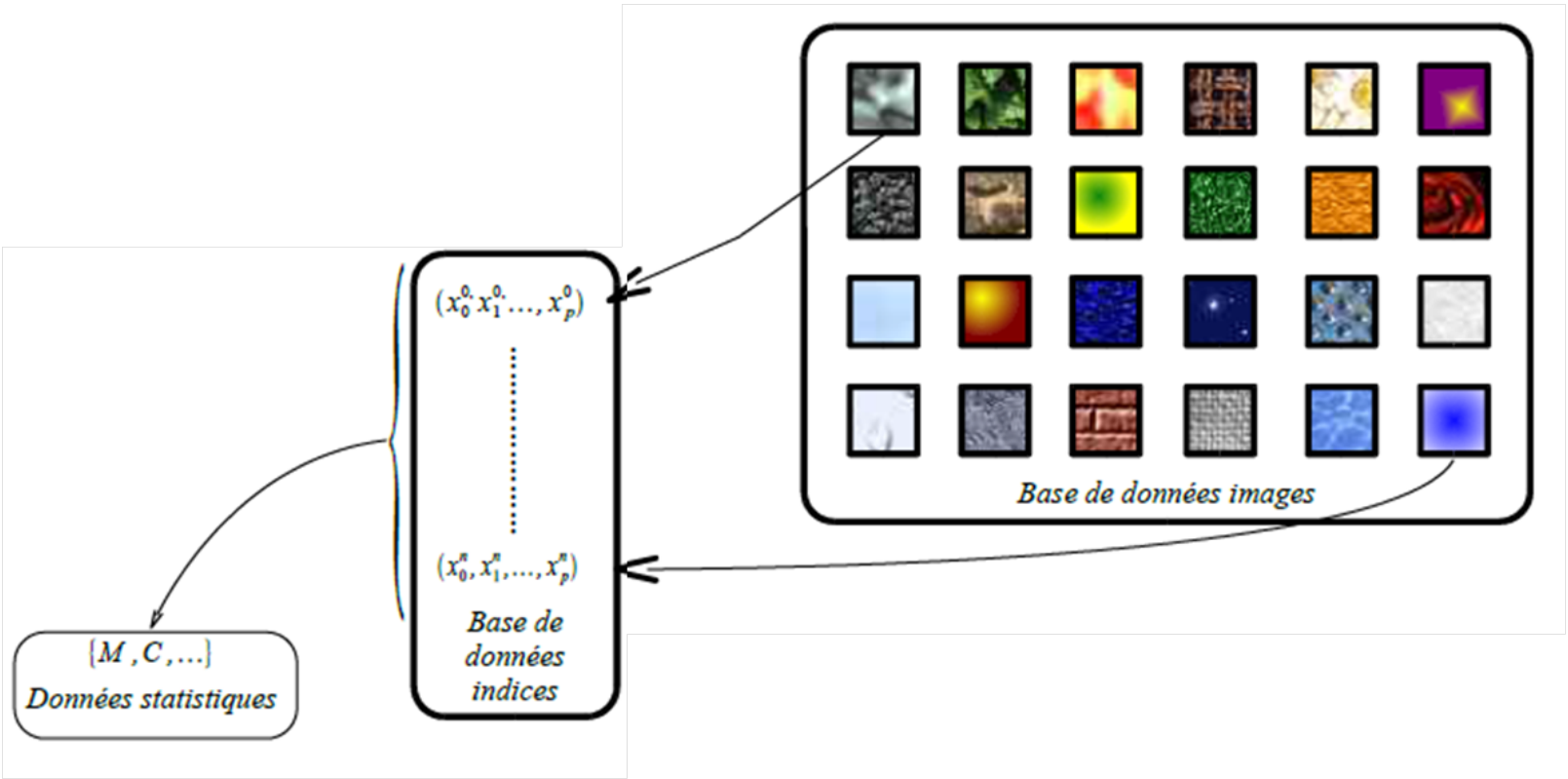


– Schéma d'indexation classique

CBIR : Principe

Hors-ligne : Indexation

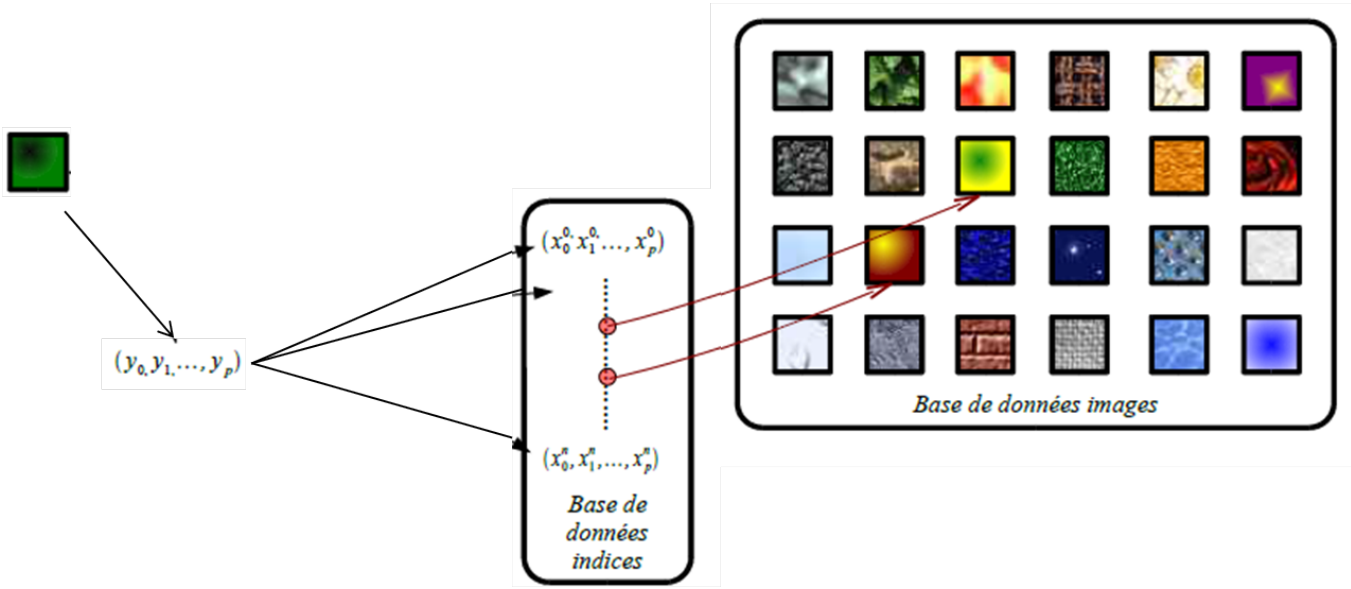
- Calcul des signatures (indices) de description pour toutes les images de la base



CBIR : Principe

En-ligne : Recherche

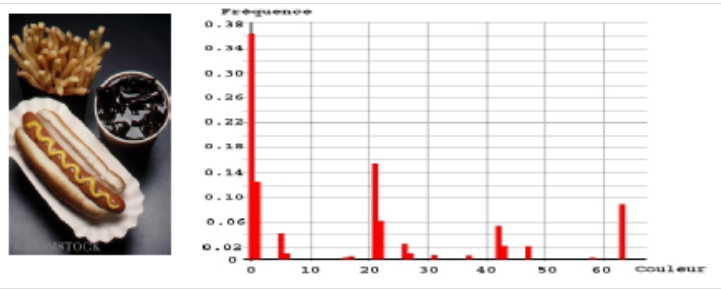
1. Calcul de **signature** pour l'image inconnue (image requête)
2. Mesure de **similarité** de la signature de l'image inconnue avec les indices de la base
3. Résultat : adresse des meilleures images au sens de la mesure de similarité



CBIR : Principe

A définir ?

- Soient $B = \{X_i = f(I_i), 1 <= i <= n\}$ et $X^{'} = f(I^{'})$ les signatures des images I_i et de l'image requête $I^{'}$ respectivement



- Mesure de similarité (exp: distance euclidienne) entre descripteurs $d(X_i, X^{'})$
- Résultat : $R = \{I_i / d(X_i, X^{'}) < \epsilon\}$

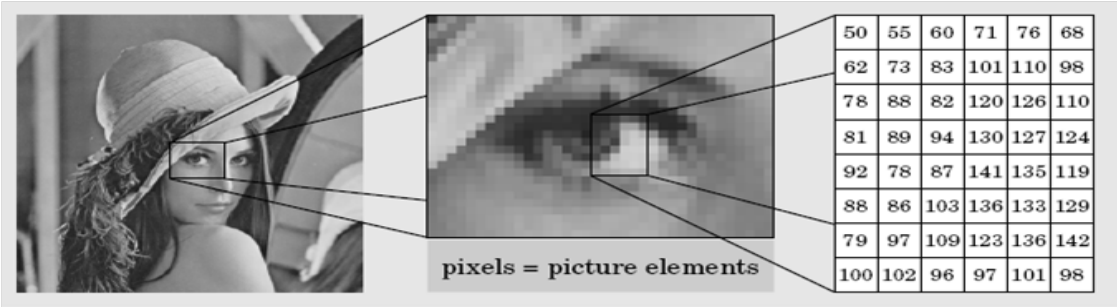
$\cdots$ ce qui nécessite

- Choix d'un espace de représentation et d'un **descripteur** (fonction)
- Calcul de **signature** de l'**image** dans cet espace
- Définition d'une **mesure de similarité (distance)** dans cet espace

Image

Image numérique

- **Image numérique** : C'est une matrice de $N * M$ pixels (picture element) correspondant à l'échantillonnage et la quantification d'un signal acquis avec un capteur



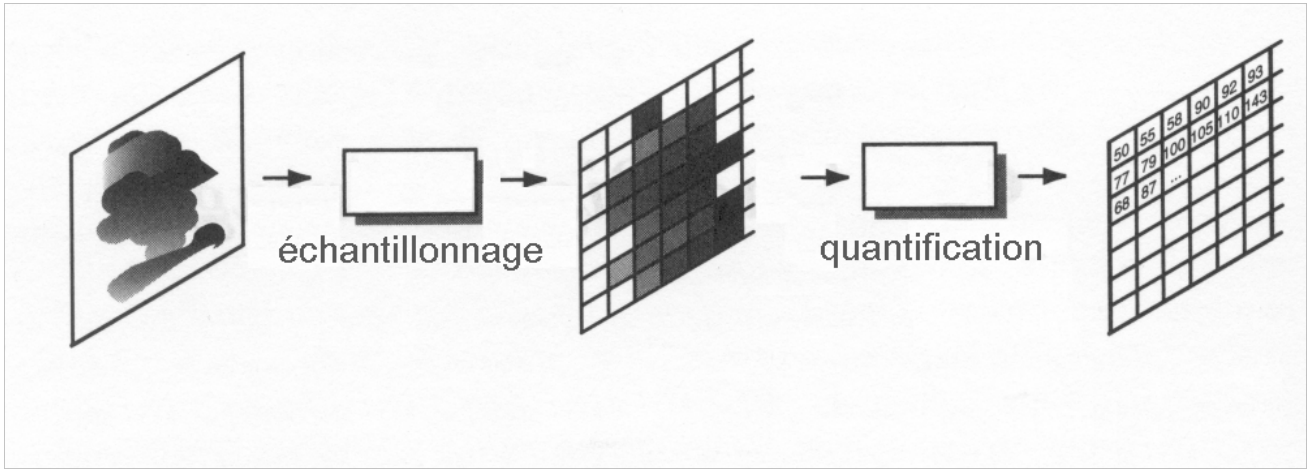
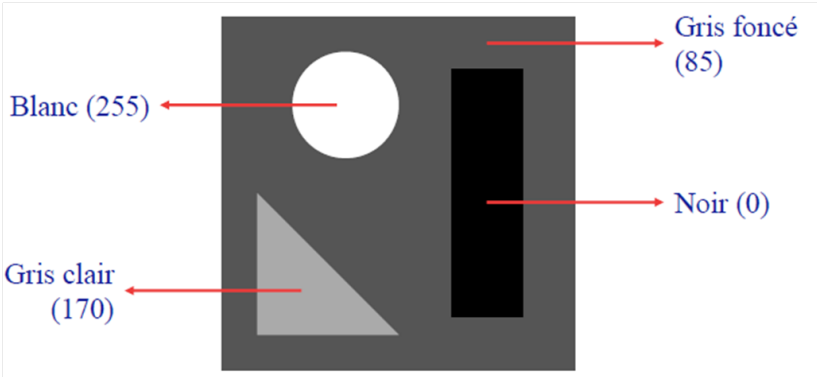


Image numérique

Format d'image

- Image à niveaux de gris (intensité ou luminance)
  - Chaque pixel est codé sur N bits, ce qui lui confère des valeurs entières comprises entre 0 (noir) et $2^N - 1$ (blanc).
- Image couleur
  - Une image couleur correspond à la synthèse additive de 3 images, rouge, vert et bleu. Chaque pixel est donc codé sur $3 \times N$ bits.



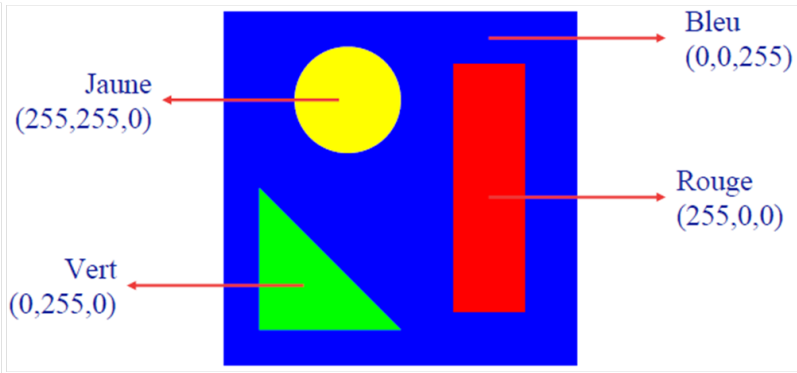


Image numérique

Image à niveaux de gris

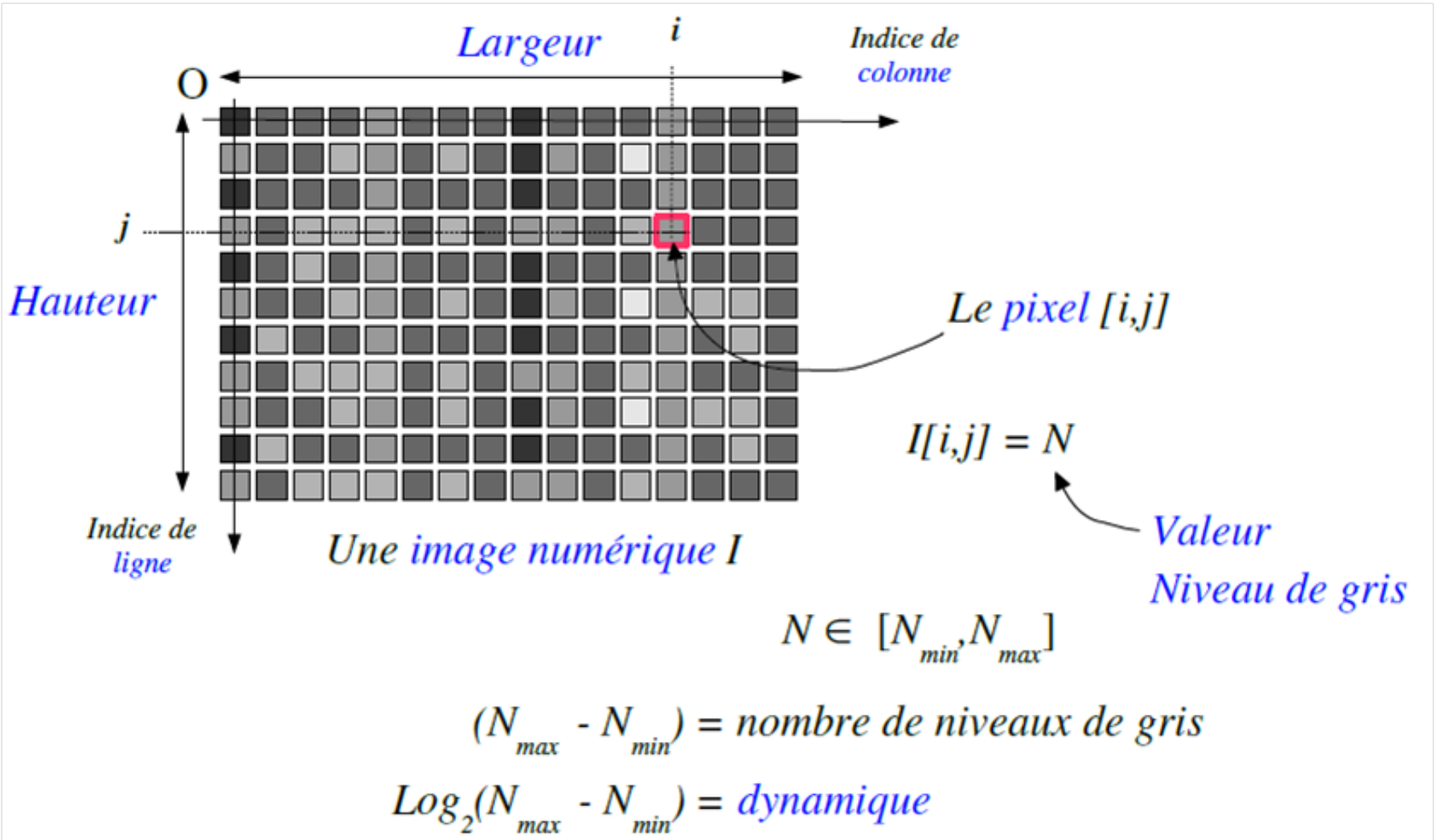
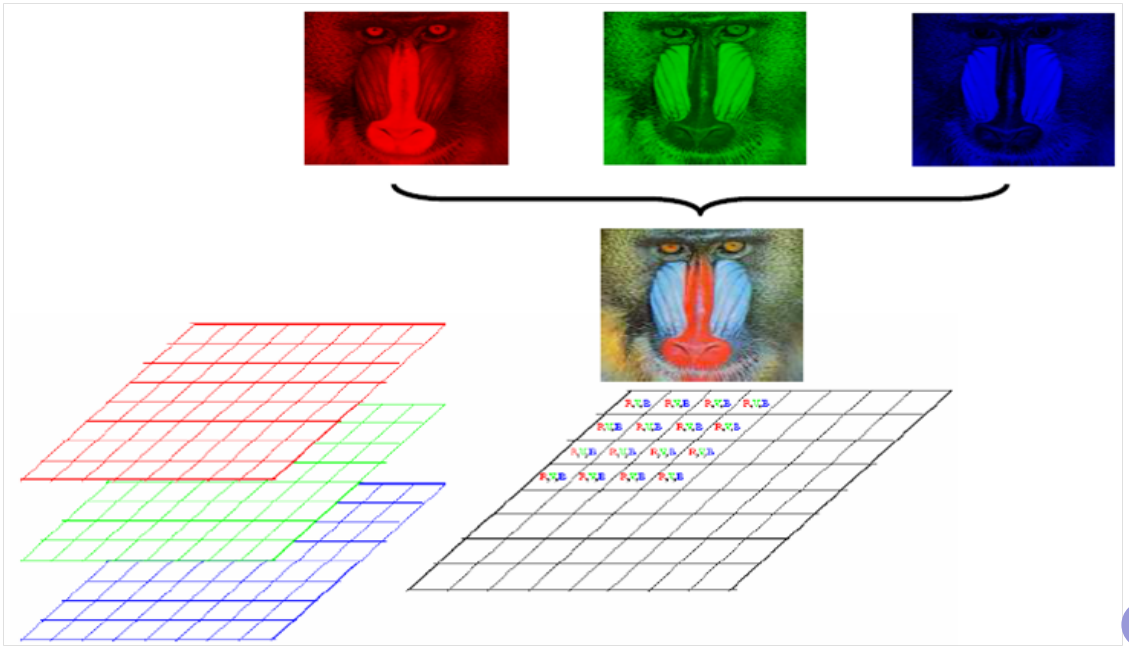


Image numérique

Image couleur

- Image couleur
  - 3 grilles de valeurs, 1 grille par composante de couleur
  - RGB : 8 bits de quantification pour chaque couleur
- $\Rightarrow$ 24 bits par pixels (pixel=élément du support du signal)



OpenCV : manipulation des images

Descripteurs d'image

Types

- Description globale de l'image : Description approximative de toute l'image
  - Considère l'image dans son ensemble
  - Caractérise l'image en utilisant des statistiques calculées sur l'image entière.
  - Une description moins fine de l'image notamment de recherche des objets.
- Description locale de l'image
  - Considère l'image comme composée d'un ensemble d'objets.
  - Détection de points d'intérêt et calculs éventuels d'invariants autour de ces points d'intérêt
- Description spécifique (essentiellement biométrie)
  - Empreintes digitales : Minuties
  - Visages : EigenFaces

Descripteurs d'image

Types de caractéristiques

- Caractéristiques globaux
  - Couleur
  - Forme
  - Texture
- Caractéristiques locales
  - Points d'intérêts
  - Régions d'intérêts
- Caractéristiques spécifiques
  - EigenFaces
  - Minuties

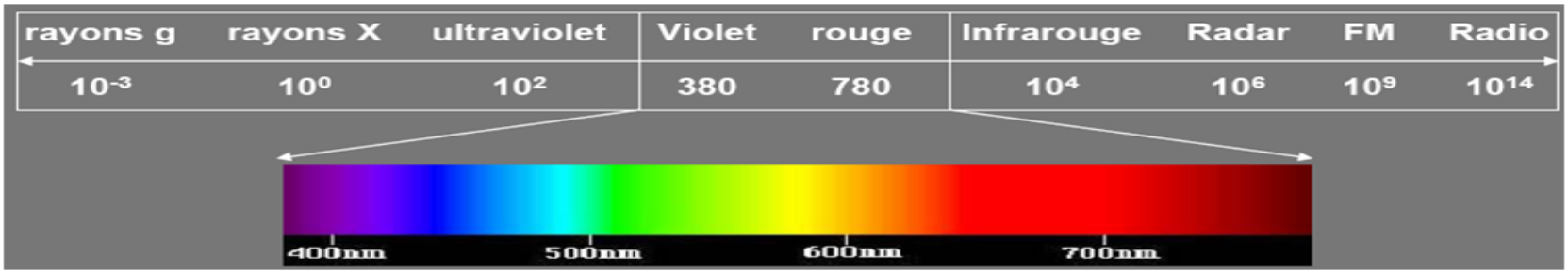
Extraction de caractéristiques (signatures) visuelles (visual features extraction) consiste en des transformations mathématiques calculées sur les pixels d'une image numérique.

Descripteurs de couleur

Indice visuel couleur

- Quelle est la couleur de cette pomme?

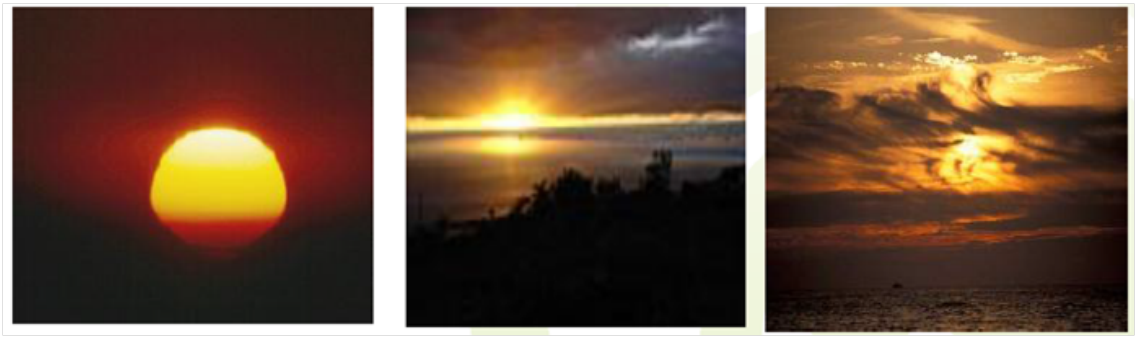
- Alors. . . Couleur est une propriété d'objet



Descripteurs de couleur

Indice visuel couleur

- **Hypothèse:** si deux images partagent des couleurs similaires alors également leur contenu peut être similaire
- **Exemple:** coucher de soleil (orange, jaune)



· · · à définir :

- **Espaces de couleur** (systèmes de représentation des couleurs)
- **Descripteurs couleur : Moments de couleur, Histogramme**

Descripteurs de couleur

Les systèmes de représentation des couleurs

- Kunt et al. a démontré qu'en combinant trois longueurs d'ondes particulières, il est possible de synthétiser presque toutes les couleurs existantes.
- En traitement d'images, les systèmes les plus couramment utilisés sont
  - Systèmes de primaires
    - le système (R,G,B)
    - le système (C,M,Y)
    - le système (X,Y,Z)
  - Systèmes luminance-chrominance :
    - les systèmes uniformes de la CIE (L*,a*,b*) et (L*,u*,v*)
    - les systèmes de télévision (Y,I,Q) et (Y,U,V)
    - les système (HSV) et (HLS)
  - Systèmes d'axes indépendants :
    - le système d'Ohta (I1,I2,I3) ou un système obtenu par ACP,

▶▶ OpenCV : conversion entre espace de couleurs

Descripteurs de couleur

Moments de couleur

- la moyenne, moment d'ordre un : $\mu \triangleq m_1 = \mathbb{E}(I)$
- la variance, moment centré d'ordre deux : $\mathrm{V}(I) \triangleq \mu_2 = \mathbb{E}[(I-\mu)^2]$
- $\cdots$, ainsi que sa racine carrée l'écart type : $\sigma \triangleq \sqrt{\mathrm{V}(I)} = \sqrt{\mu_2}$
- le coefficient d'asymétrie, moment centré réduit d'ordre trois :

$$\gamma_1 \triangleq \beta_1 = \mathbb{E}\left[\left(\frac{I-\mu}{\sigma}\right)^3\right]$$

- le kurtosis non normalisé, moment centré réduit d'ordre quatre :

$$\beta_2 = \mathbb{E}\left[\left(\frac{I-\mu}{\sigma}\right)^4\right]$$

Descripteurs de couleur

Moments de couleur : couleur moyenne

- La valeur moyenne (dans l'espace RGB): somme des valeurs RGB de tous les pixels, normaliser par le nombre de pixels

$$R_{avg} = \frac{1}{N \times M} \sum_{i=0}^{N} \sum_{j=0}^{M} R(i,j), G_{avg} = \frac{1}{N \times M} \sum_{i=0}^{N} \sum_{j=0}^{M} G(i,j)$$

$$B_{avg} = \frac{1}{N \times M} \sum_{i=0}^{N} \sum_{j=0}^{M} B(i,j)$$

- Comparaison de deux images x et y par la couleur moyenne en utilisant la distance euclidienne

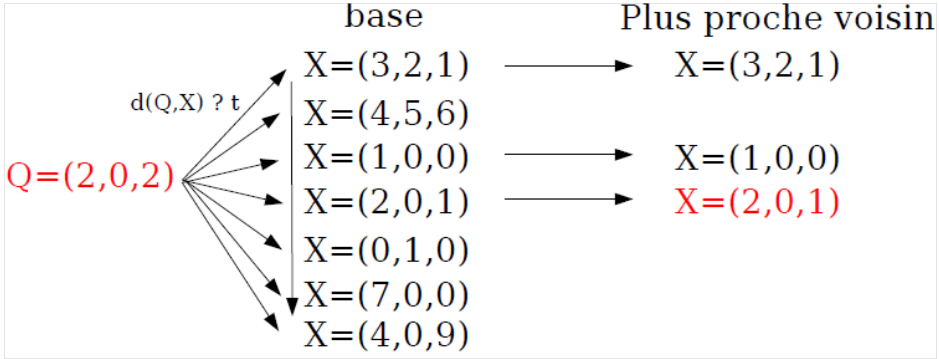
$$d^2_{avg}(x,y) = (R_{avg}x - R_{avg}y)^2 + (G_{avg}x - G_{avg}y)^2 + (B_{avg}x - B_{avg}y)^2$$

▸▸ OpenCV : moments de couleur (couleur moyenne)

Descripteurs de couleur

CBIR : Moments de couleur

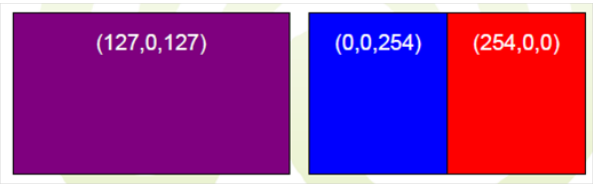
- Q signature (moment de couleur) d'une image requête I_Q
- $B = \{X_i \ (1 < i < N)\}$ Base de signatures des images de la base
- Quelle(s) est (sont) l'image(s) de la base la (les) plus similaire(s) à l'image requête ?
  - Quelles sont les images similaires à Q ? sont Xi tels que $d(Q, X_i) \leq \varepsilon$
    - ε : seuil déterminer par l'utilisateur



Descripteurs de couleur

CBIR : Moments de couleur

- Limites :
  - Mesure de similarité non précise
  - Par exemple, l'image violette et l'image rouge-bleu sont les même selon la moyenne de couleur



- Mais
  - Rapide et facile à calculer et comparer
  - Meilleur pour l'utiliser comme un filtre: exclure images
  - Couleur dominante influence la moyenne de couleur

Descripteurs de couleur

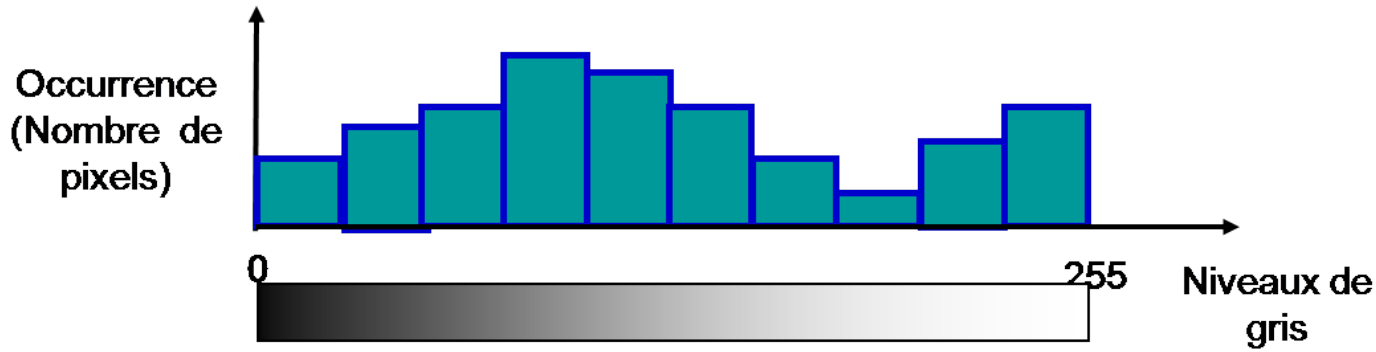
CBIR : Moments de couleur

- Exemple : rechercher des images surtout jaunes : exclure toutes les images avec des moyennes rouge, bleu ou verte

Descripteurs de couleur

Histogramme

- L'histogramme est une fonction $Hist(i)$ permettant de donner la fréquence d'apparition des différents niveaux de gris(couleur) i qui composent l'image
  - En abscisse on représente les niveaux de gris (couleur) et en ordonnée leurs fréquences d'apparition
  - L'histogramme des niveaux de gris (couleur) nous informe sur la concentration de l'image

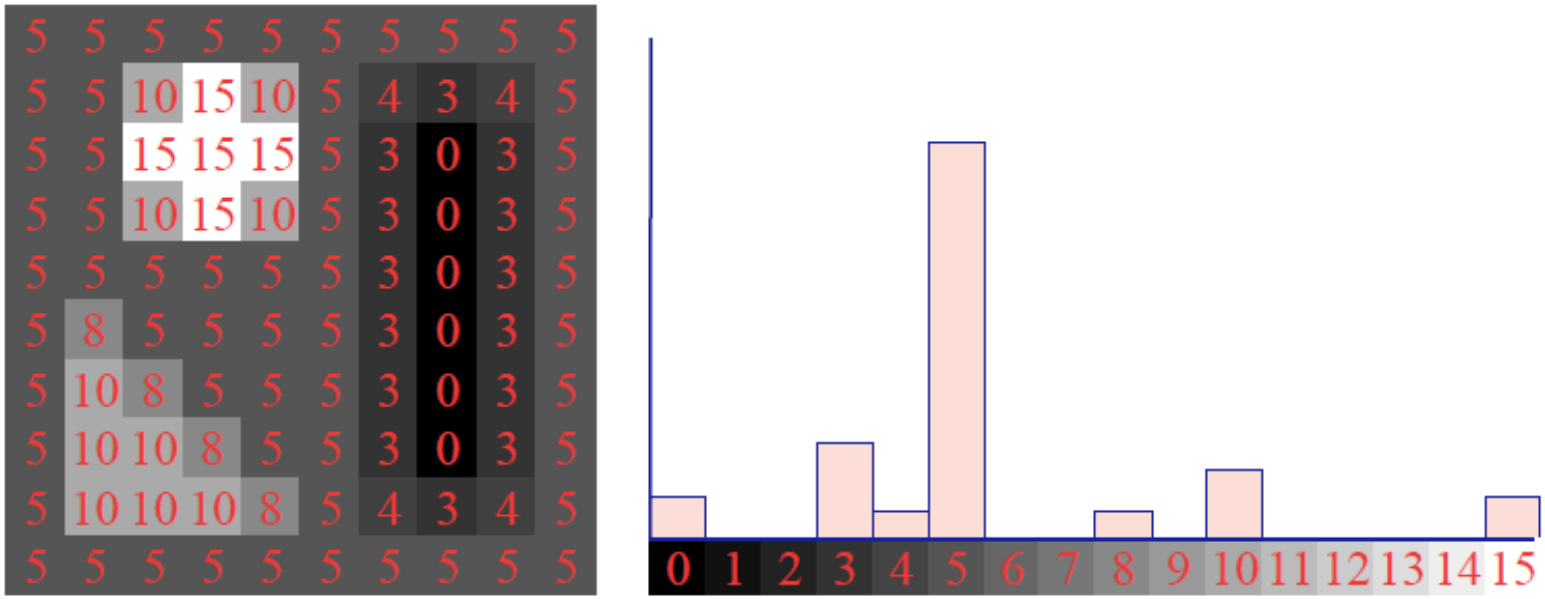


- Pour une image couleur, il y a un histogramme par composante

Descripteurs de couleur

Histogramme

- Fonction discrète $[0, L-1]$
  - $Hist(i) = n$
  - $i =$ ton de gris
  - $n =$ nombre de pixel de ton i

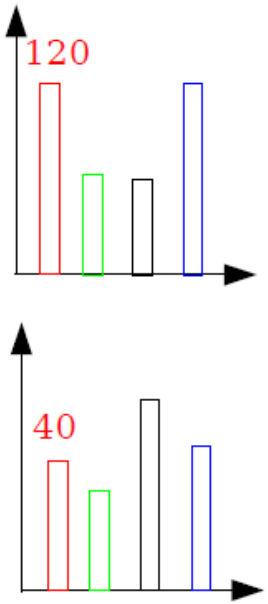


Descripteurs de couleur

CBIR : Histogramme

- L'intersection d'histogrammes

$$d(X,X') = 1 - \frac{\min(x_1, x'_1) + \min(x_2, x'_2) + \ldots + \min(x_D, x'_D)}{\min(x_1 + x_2 + \ldots + x_D, x'_1 + x'_2 + \ldots + x'_D)}$$



Min(120,40)=40 → Au moins 40 pixels rouges dans chacune des deux images

\+ Au moins 35 pixels verts dans chacune des deux images

\+ Au moins 34 pixels noirs dans chacune des deux images

\+ Au moins 50 pixels noirs dans chacune des deux images

159 pixels ayant une couleur commune dans les deux images

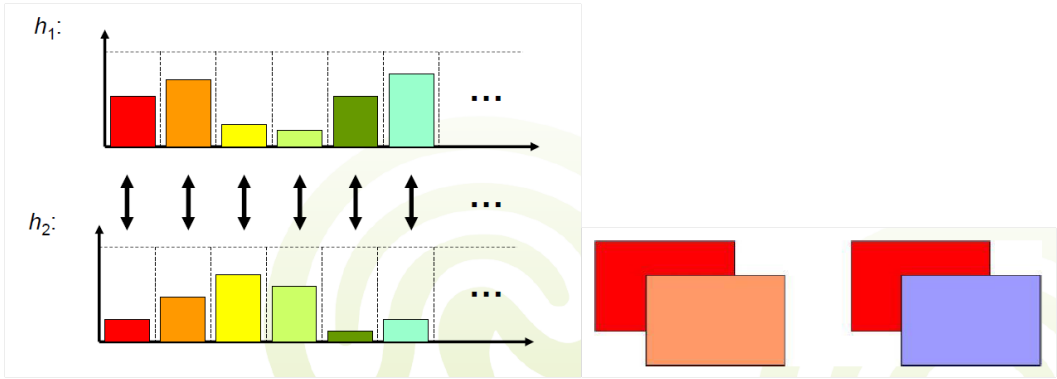
OpenCV : histogramme

Descripteurs de couleur

CBIR : Histogramme

- Distance du Minkowski
  - Soient h1 et h2 deux histogrammes
  - La distance de Minkowski avec le paramètre r:

$$d_r = \sum_{i \in C} | h_1(i) - h_2(i) |^r$$



  - La distance entre une image rouge et une image rouge vif est la même que entre une image rouge et image bleu
  - Limité dans le cas de changements de couleur parce que toutes les colonnes sont comparées individuellement

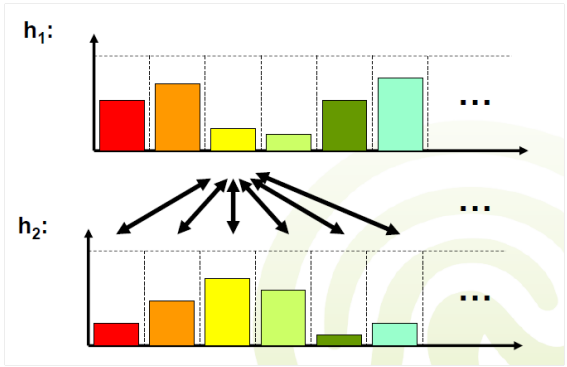
Descripteurs de couleur

CBIR : Histogramme

- Distance quadratique
  - Évaluer la relation entre les différentes couleurs
  - Soit A une matrice qui exprime la similarité des paires entre la couleur i et la couleur j ($a_{i,i} = 1$ et $a_{i,j} = a_{j,i}$):

$$d_A(h_1, h_2) = (h_1 - h_2)^T \cdot A \cdot (h_1 - h_2)$$

$$= \sum_{j \in C}^{k} \sum_{i \in C}^{k} (h_1(i) - h_2(i))((h_1(j) - h_2(j))$$



- Autres distances : la distance de Mahalanobis, Chi2

Descripteurs de couleur

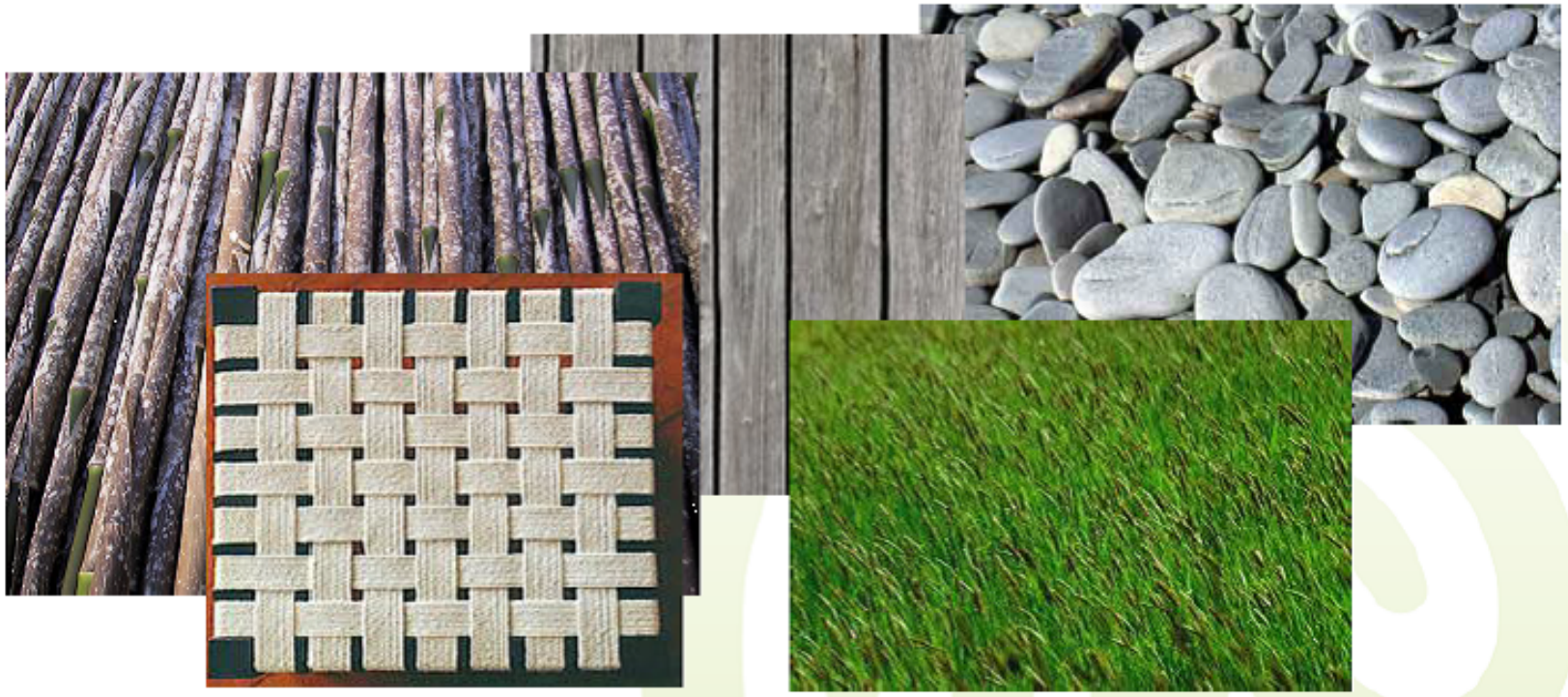
CBIR : Histogramme

- Problèmes
  - choix de la représentation de la couleur
  - distance entre couleurs
- Limites
  - Indice visuel insuffisant si utilisé seul, car insuffisamment discriminant : pomme rouge vs Ferrari
- Avantages des histogrammes
  - Robuste à certaines transformations géométriques de l'image
- Remarque : une image en noir et blanc suffit à un humain pour effectuer la tâche demandée!

Descripteurs de texture

Textures : Quelle définition

- **En vision par ordinateur**, il n y a pas de définition complètement satisfaisante : Zone homogène en un certain sens, assemblage d'entités élémentaires formant un tout.



Descripteurs de texture

Comment analyser texture ?

Méthodes de description de la texture

- Méthodes statistiques
  - **Matrice de coocurrence (Mesures de Haralick)**
  - **Histogramme**
- Méthodes à base de modèle
  - Décomposition de Wold
  - Modèles Fractals
  - Modèles AR (Autoregressive Models)
- Méthodes fréquentielles (traitement du signal)
  - **Fourier**
  - Gabor
  - Ondelettes

Il existe plusieurs méthodes pour analyser la texture, mais le plus difficile est de trouver une bonne représentation(paramètres) pour chaque texture

Descripteurs de texture

Matrice de coocurrence (Mesures de Haralick)

- L'idée de cette méthode est d'identifier les répétitions de niveaux de gris selon une distance (pas) et une direction
- Matrice de coocurrence : matrice de taille $Ng \times Ng$
  - Ng étant le nombre de niveaux de gris de l'image (256x256)
  - On réduit souvent a des tailles 8x8, 16x16 ou 32x32
- Pour un voisinage (dx, dy), la matrice de cooccurrence $M(dx, dy)$ est donnée par :

$$M\,[dx, dy\,](u, v) =$$

$$\frac{1}{(N_x dx)(N_y dy)} \sum_{i,j} 1\,[I(i,j) = u \& I(i+dx, j+dy) = v\,]$$

  - (Nx, Ny) : taille de l'image
  - (u, v) : niveaux de gris de l'image (valeur quantifiée)

Descripteurs de texture

Matrice de coocurrence (Mesures de Haralick)

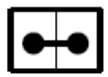
- Plusieurs matrices, pour chaque distance (pas) et direction
  - Distance : $1, 2, 3(, 4, \cdots, K)$
  - Direction : $0, 45, 90, 135, (^\circ)$
- Temps de calcul de ces matrices est assez long
- Soit l'image I définie par:

1	4	4	3
4	2	3	2
1	2	1	4
1	2	2	3

Image

	1	2	3	4
1	?	?	?	?
2	?	?	?	?
3	?	?	?	?
4	?	?	?	?

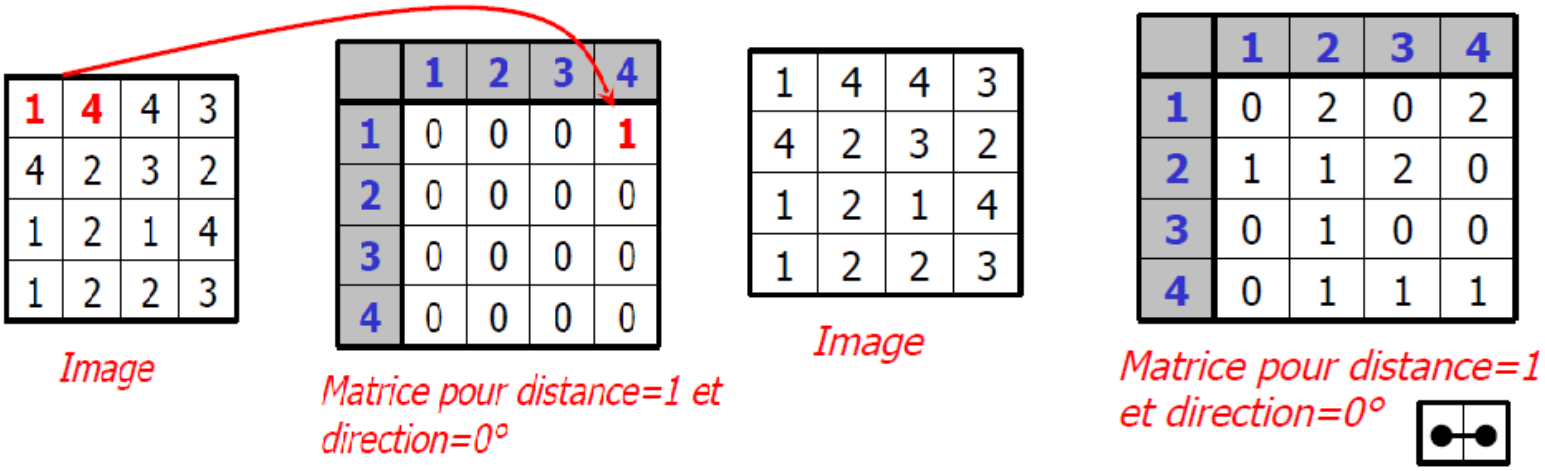
Matrice pour distance=1 et direction=0°



Descripteurs de texture

Matrice de coocurrence (Mesures de Haralick)

- On parcours l'image et pour chaque couple de pixels formé avec la distance et la direction données, on incrémente la matrice des cooccurrences de 1
- ... alors la matrice de cooccurrence (non normalisée) est:



- Le 2 de la matrice de cooccurrence (ligne 1 et colonne 4) signifie que l'on trouve deux fois un pixel de valeur 1 de distance 1 et de direction 0 d'un pixel de valeur 4.

Descripteurs de texture

Matrice de coocurrence (Mesures de Haralick)

- À partir de cette matrice de cooccurrence, il est possible de définir plusieurs descripteurs (Mesures de Haralick), tels que ceux répertoriés dans cette table:

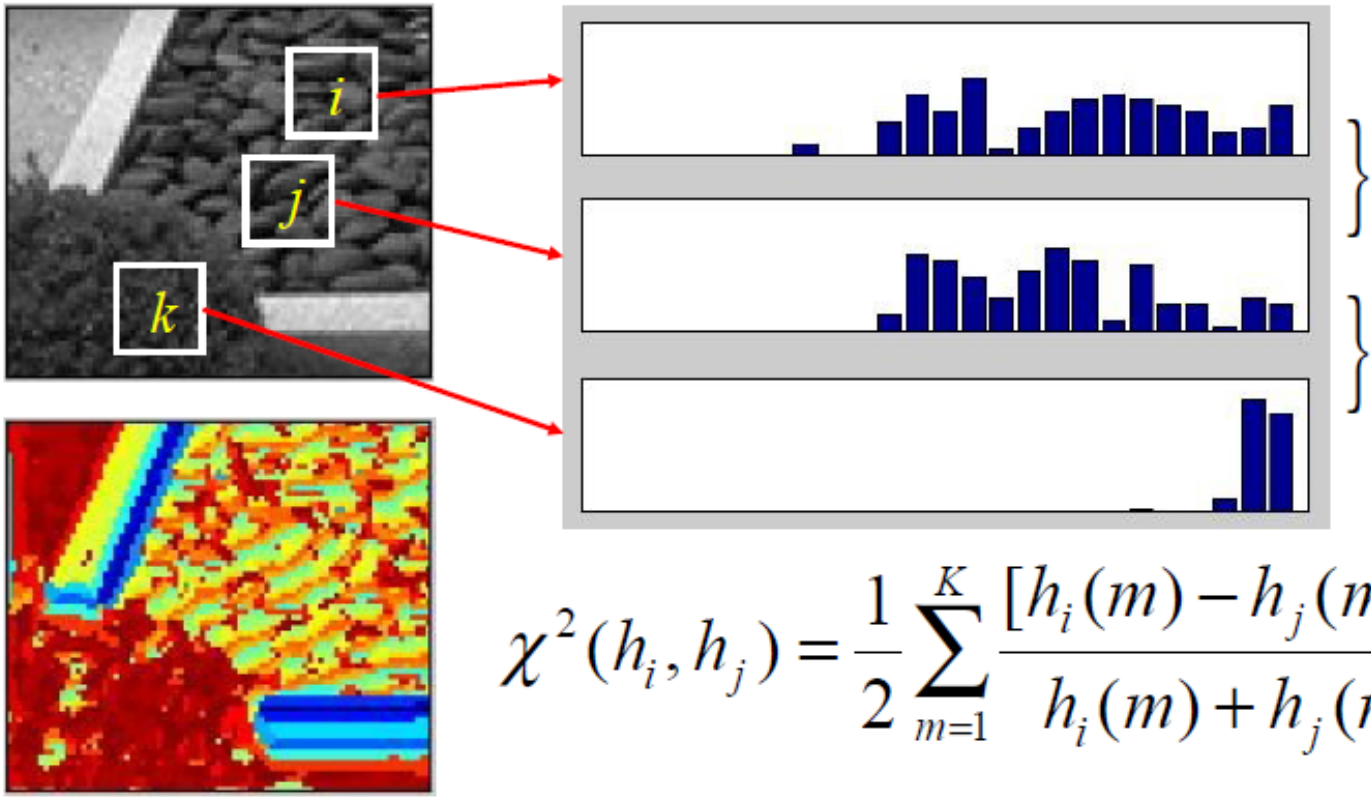
Opérateur	Formulation
Maximum	$\max_{ij}(C_{ij})$
Différence d'ordre k	$\sum_i \sum_j c_{ij} (i-j)^k$
Entropie	$\sum_i \sum_j c_{ij} log(c_{ij})$
Uniformité	$\sum_i \sum_j c_{ij}^2$

▸▸ OpenCV : matrice de coocurrence (Mesures de Haralick)

Descripteurs de texture

Histogramme

- Statistiques du premier ordre (histogramme)
  - Distance du Chi2 entre histogrammes de textures



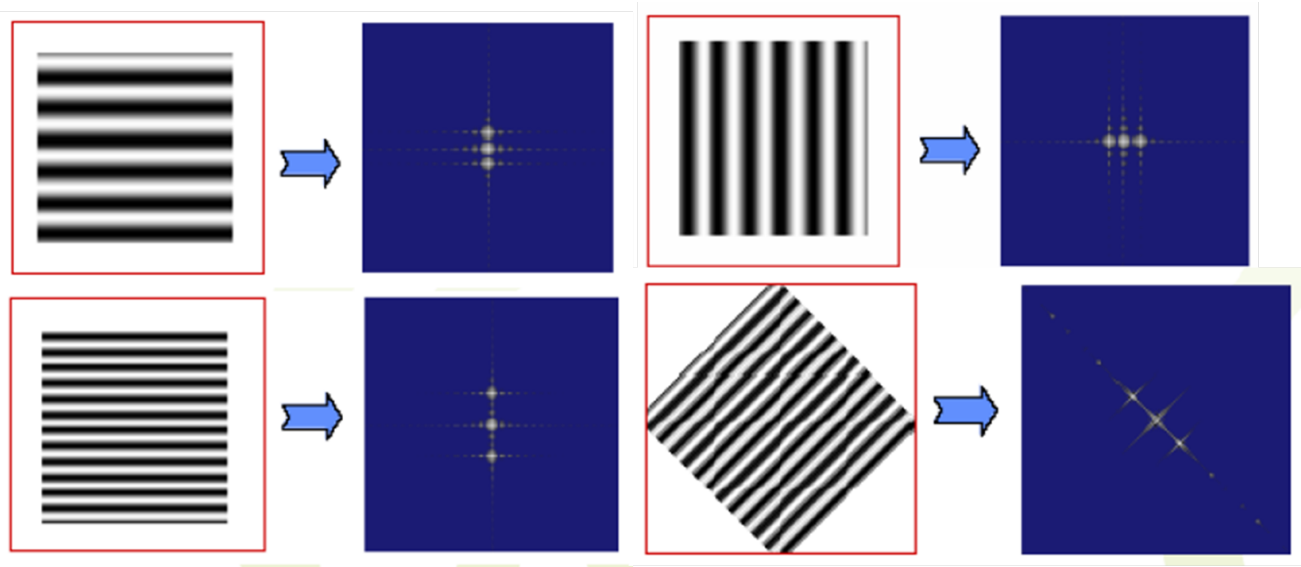
$$\chi^2(h_i, h_j) = \frac{1}{2}\sum_{m=1}^{K}\frac{[h_i(m) - h_j(m)]^2}{h_i(m) + h_j(m)}$$

Descripteurs de texture

Approche spectrale

- Transformée de Fourier discrète
  - En traitement d'images, on utilise la transformation de Fourier à deux dimensions, sa définition discrète est :

$$\mathrm{F}(u,v) = \sum_{m=0}^{\mathrm{M}-1} \sum_{n=0}^{\mathrm{N}-1} I(m,n) \cdot \mathrm{e}^{-2i\pi\left(\frac{um}{\mathrm{M}}+\frac{vn}{\mathrm{N}}\right)}$$

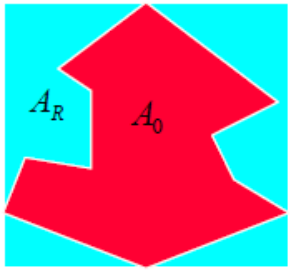


OpenCV : Transformée de Fourier Discrète

Descripteurs de forme

- Caractérisation des objets contenus dans l'image : représentation de composantes
  - Caractéristiques internes : description de la région occupée par l'objet (surface, ...)
  - Caractéristiques externes : description du contour de l'objet représenté par le périmètre, circularité, rectangularité, ...
- Cette description est invariante en rotation et translation.

elongation	$\frac{W}{H}$ avec H : hauteur, W : largeur
circularité	$\frac{T^2}{4\pi A}$ avec T: périmetre, A : surface
rectangularité	$\frac{A0}{Ar}$ avec $A0$: surface de la région, Ar : surface de rectangle



- Décrire les formes nécessite une identification préalable de régions
  - Segmentation de l'image
  - Détection de leurs contours

Image numérique

OpenCV : manipulation des images

- La classe **MAT**
  - Pour lire une image utiliser la fonction **imread**

```
Mat imread(const string& filename, int flags=1)

Paramètres :

    filename - Name of file to be loaded.

    flags - Specifies color type of the loaded image:

         >0 the loaded image is forced to be a 3-channel
    color image

        =0 the loaded image is forced to be grayscale

        <0 the loaded image will be loaded as-is.
```

  - Les objets Mat A, B, C, qui possèdent des entêtes propres à chacun qui pointent sur les mêmes données (sur la même matrice)

```
Mat A, C; // création des entêtes
A = imread(argv[1], CV_LOAD_IMAGE_COLOR); // chargement
d'une image
Mat B(A); // appel du constructeur de copie
C = A; // affectation
```

Image numérique

OpenCV : manipulation des images

- La classe **MAT**
  - Si l'on cherche à préserver les données de l'objet Mat original, il faut manipuler les données d'une copie. Pour cela, il existe les méthodes **clone()** et **copyTo()**

```
Mat F = A.clone();
Mat G;
A.copyTo(G);
```

  - Accès aux pixels d'une image
    - Pour accéder à la valeur d'un pixel(x, y), on utilisera la fonction **at()** de la classe Mat ou la fonction **cvGet2D**. La fonction **at()** nécessite que l'on lui indique le prototype des données à extraire

```
//La méthode générique at :

Template <typename _Tp> _Tp& Mat::at(int i, int j)
Template <typename _Tp> _Tp& Mat::at(Point pt)

Paramètres :
    i - The row index (start at 0)
    j - The column index (start at 0)
    pt - The element position specified as Point(j,i)
```

Image numérique

OpenCV : manipulation des images

- La classe **MAT**
  - L'objet **Mat** obtenu à partir d'une image couleur possède trois canaux, le Bleu, le Vert et le Rouge. L'ordre a son importance. Par défaut dans OpenCV, le bleu est chargé dans le premier canal, le vert dans le second et le rouge dans le troisième

```
Vec3b intensity = bgrMat.at<Vec3b>(x, y);
uchar blue = intensity.val[0];
uchar green = intensity.val[1];
uchar  red = intensity.val[2];
```

  - **Exemple** : exemple qui inverse une image couleur, on traite donc directement des triplets RGB représentés sous forme de **Vec3b**

```
// invert the image
for(int i=0;i<height;i++)
    for(int j=0;j<width;j++)
    img.at<Vec3b>(i,j)=Vec3b(255,255,255)-img.at<Vec3b>(i,j);
```

Image numérique

OpenCV : manipulation des images

- La classe **MAT**
  - Méthode par découpage en plan : cette approche consiste simplement à découper une image à *n* canaux en *n* image à 1 canal, traiter chaque canal, puis les réassembler
    - La séparation des canaux se fait grâce à fonction **split**

```
// split the image into separate color planes
vector<Mat> planes;
split(img, planes);
// invert the 3 planes
for(int c=0;c<planes.size();c++)
    for(int i=0;i<height;i++)
        for(int j=0;j<width;j++)
    planes[c].at<uchar>(i,j)=255-planes[c].at<uchar>(i,j);
// now merge the results back
merge(planes, img);
```

  - La fonction **merge** assemble plusieurs images à 1 canal contenues dans un vecteur de *Mat* en une unique image

Image numérique

OpenCV : manipulation des images

- La classe **MAT**
  - Les images à niveau de gris ne possède qu'un canal ce qui facilite leur manipulation. Le type de données stocké sera généralement des **uchar** (8 bit non signé) ou des **float** (32 bit)
  - **Exemples:**
    - Pour créer un objet **Mat grey** ayant la même taille que **bgrMat**, mais qui ne possède qu'un canal, on utilisera la méthode **create()** de la classe **Mat**

```
Mat grey ;
grey.create(bgrMat.size(),CV_8U);
```

    - Pour obtenir une image en niveaux de gris donc sur un canal de 8 bits, écrire :**Mat img = imread(filename, 0);**, ou utiliser la fonction **cvtColor(src, src_gray,** CV_RGB2GRAY);

Image numérique

OpenCV : manipulation des images

- Affichage d'une image : l'affichage d'une image se réalise en deux temps (+ un troisième optionnel)
  - Création d'une fenêtre nommée avec **namedWindow**
  - Affichage de l'image avec **imshow**

```
void namedWindow(const string& winname, int flags)

void imshow(const string& winname, const Mat& image)
```

- Enregistrement d'une image
  - L'enregistrement se fait avec **imwrite**

```
bool imwrite(const string& filename, const Mat& img, const
vector<int>& params=vector<int>())

params (optional) - The format-specific save parameters,
encoded as pairs paramId_1, paramValue_1, paramId_2,
paramValue_2, ... . The following parameters are currently
supported:
      • In the case of JPEG it can be a quality
        (CV_IMWRITE_JPEG_QUALITY ), from 0 to 100 (the higher is the
        better), 95 by default.
      • In the case of PNG it can be the compression level
        (CV_IMWRITE_PNG_COMPRESSION ), from 0 to 9 (the higher value
        means smaller size and longer compression time), 3 by default.
      • In the case of PPM, PGM or PBM it can a binary format flag
        (CV_IMWRITE_PXM_BINARY ), 0 or 1, 1 by default.
```

Image numérique

OpenCV : manipulation des images

```
#include <iostream>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
using namespace std;
using namespace cv;
int main(int argc, char *argv[]) {
    // load an image
    Mat img=imread("camera.png",0); if(img.empty()){
        printf("Could not load image file\n");
        exit(0);
    }
    // get the image data
    int height = img.rows;
    int width = img.cols;
    printf("Processing a %dx%d image\n",height,width);
    // invert the image
    for(int i=0;i<height;i++)
        for(int j=0;j<width;j++)
            img.at<uchar>(i,j)=255-img.at<uchar>(i,j);
    // create a window
    namedWindow("mainWin", CV_WINDOW_AUTOSIZE);
    // show the image
    imshow("mainWin", img );
    // wait for a key
    waitKey(0);
    return 0;
}
```

- On peut aussi lire l'image par la fonction **IplImage* cvLoadImage (const char* filename, int iscolor=*CV_LOAD_IMAGE_COLOR*)**

Retour

Descripteurs de couleur

OpenCV : conversion entre espace de couleurs

- Conversion entre espace de couleurs
  - Fonction : cvtColor(imagesrc, imagedest, code)
  - Codes : $CV\_ < colorspace > 2 < colorspace >$
  - Exemples: **CV_BGR2HSV, CV_BGR2LUV, CV_BGR2GRAY, ...**

- **Exemples:**
  - Conversion de RGB en LUV
    $CvtColor(imgsrc, imgdest, CV\_RGB2LUV)$;
  - Conversion de RGB en HSV
    $CvtColor(imgsrc, imgdest, CV\_RGB2HSV)$;

Retour

Descripteurs de couleur

OpenCV : moments de couleur (couleur moyenne)

- Exemple :

```
Double R_avg=0, G_avg=0, B_avg=0 ;

Vec3b color ;

for( int row = 0 ; row < bgrMat.rows; row ++) {
      for (int col = 0; col < bgrMat.cols; col ++) {
                color = bgrMat.at<Vec3b>(row, col);
                 R_avg+ = color[0];
                 G_avg+= color[1];
                 B_avg+= color[2];
       }
}

R_avg=R_avg/ bgrMat.rows*bgrMat.cols ;
G_avg=G_avg/ bgrMat.rows*bgrMat.cols ;
B_avg=B_avg/ bgrMat.rows*bgrMat.cols ;
```

Retour

Descripteurs de couleur

OpenCV : histogramme

- L'accès aux valeurs de pixels peuvent être utilisés pour calculer les valeurs de l'histogramme.
- Ou utiliser la fonction de la bibliothèques Opencv.

```
void calcHist(const Mat* images, int nimages, const int*
channels, InputArray mask, OutputArray hist, int dims,
const int* histSize, const float** ranges, bool
uniform=true, bool accumulate=false )
```

- On peut alors comparer les histogrammes couleurs avec la fonction **double compareHist(const SparseMat H1, const SparseMat H2, int method)**

Descripteurs de couleur

OpenCV : histogramme

- **Exemple : fonctions prédéfinies**

```
/// Calculate the histograms for the HSV images
calcHist( &hsv_base, 1, channels, Mat(), hist_base, 2, histSize, ranges, true, false );
normalize( hist_base, hist_base, 0, 1, NORM_MINMAX, -1, Mat() );

calcHist( &hsv_test1, 1, channels, Mat(), hist_test1, 2, histSize, ranges, true, false );
normalize( hist_test1, hist_test1, 0, 1, NORM_MINMAX, -1, Mat() );

/// Apply the histogram comparison methods(0, 1, 2, 3 : méthode de comparaisons)

   double base_test1_0 = compareHist( hist_base, hist_test1, 0);
   double base_test1_1 = compareHist( hist_base, hist_test1, 1 );
   double base_test1_2= compareHist( hist_base, hist_test1, 2 );
   double base_test1_3= compareHist( hist_base, hist_test1, 3 );
```

Descripteurs de couleur

OpenCV : histogramme

- **Exemple**

```
void computeColorHist(const Mat& image, MatND& hist, int N)
{
	const int histSize[] = {N, N, N};

	hist.create(3, histSize, CV_32F);

	hist = Scalar(0);

	CV_Assert(image.type() == CV_8UC3);
	MatConstIterator_<Vec3b> it = image.begin<Vec3b>(),it_end = image.end<Vec3b>();

	for( ; it != it_end; ++it ){
		const Vec3b& pix = *it;
		hist.at<float>(pix[0]*N/256, pix[1]*N/256, pix[2]*N/256) += 1.f;
	}
}
```

Descripteurs de couleur

OpenCV : histogramme

- **Exemple**

```
Vec3b color;

    for(x = 0; x < img.rows; x++)
    {
        for(y = 0; y < img.cols; y++)
        {
            color = img.at<Vec3b>(x,y); //cvGet2D(img, x, y);

            histo[colorIndex][color.val[colorIndex]]++;
        }
    }
```

Retour

Descripteurs de texture

OpenCV : matrice de coocurrence (Mesures de Haralick)

- **Exemple**

```
for(x = 0; x < imgGris.rows; x++)
   {
       for(y = 0; y < imgGris.cols; y++)
       {
           if(x+dx >= 0 && x+dx < imgGris.rows && y+dy >= 0 && y+dy < imgGris.cols)
           {
               color1 = imgGris.at<Vec3b>(x, y);
               color2 = imgGris.at<Vec3b>(x+dx, y+dy);
               mat[color1.val[0]][color2.val[0]]++;
           }
       }
   }
```

Descripteurs de texture

OpenCV : matrice de coocurrence (Mesures de Haralick)

- **Exemple**

```
// Retourne l'entropie à partir d'une matrices de cooccurence d'une image
double getEntropieMatCooccurence(int ** mat, int channel)
{
    int x, y;
    double entropie = 0.;

    for(x = 0; x < channel; x++)          // Calcul de l'entropie
        for(y = 0; y < channel; y++)
            entropie -= mat[x][y] * log10(mat[x][y]+1);       // <==== Attention rajout de 1+ dans le log...

    return entropie;
}
```

Retour

Descripteurs de texture

OpenCV : Transformée de Fourier Discrète

- DFT : l'analyseur de spectre (transformation de Fourier rapide)
  - Elle permet de passer d'une représentation spatiale à une représentation fréquentielle de l'image.
- La DFT est plus rapide quand la hauteur ou la largeur de l'image est un multiple de 2, 3 ou encore 5.
- La fonction **int getOptimalDFTSize(int vecsize)** permet d'obtenir la hauteur ou la largeur préférentielle qui optimise le calcul de la DFT
- Il faut étendre l'image à ces mesures, grâce à la fonction **copyMakeBorder**
- Utiliser la fonction **dft** pour calculer la DFT
- **Exemple :** Soit padded, l'image correspondant à l'image d'origine dont la taille a été optimisée, soient planes, un tableau de 2 images, soit complexI, la Mat qui accueillera la DFT

Descripteurs de texture

OpenCV : Transformée de Fourier Discrète

```
Mat I = imread(img, CV_LOAD_IMAGE_GRAYSCALE);

    if( I.empty())

        return -1;

    Mat padded;

    int m = getOptimalDFTSize( I.rows );

    int n = getOptimalDFTSize( I.cols );

copyMakeBorder(I,padded,0,m - I.rows,0,n - I.cols, BORDER_CONSTANT,
Scalar::all(0));

Mat planes[]={Mat_<float>(padded),Mat::zeros(padded.size(),
CV_32F)};

    Mat complexI;

    merge(planes, 2, complexI);

    Mat imgTF;

dft(complexI, imgTF);
```

Retour