



# Indexation

, Extraction et Recherche d'Informations dans les Données Multimédias

A. ELHASSOUNY

GL  
ENSIAS

Master B2dS





- 1 Indexation des données multimédias
  - 1 Indexation des données multimédias : Introduction
  - 2 Description et indexation des images (CBIR 1)
  - 3 Description et indexation des images (CBIR 2)
  - 4 Description et indexation des vidéos (CBVR)
  - 5 Structures d'index 1, 2 et 3
- 2 Introduction to Deep learning in computer vision
- 3 Annexe : Evaluation de système de recherche de données multimédia

# Indexation des données multimédias

## 1 Introduction

- Contexte
- Bases de Données MultiMédia (MMDB)
- Domaines d'application
- Indexation
- Outil OpenCV

# Contexte

## Objectif



- Les techniques présentées ci-après, dite d'indexation, se proposent d'attacher à une image ou à une video un ensemble de descripteurs de leur contenu, dans le but de mesurer la ressemblance avec les descripteurs correspondant à la requête.

# Contexte

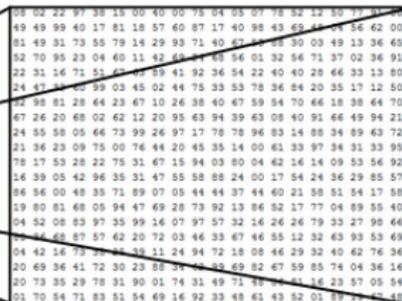
## Objectif



- Les techniques présentées ci-après, dite d'indexation, se proposent d'attacher à une image ou à une video un ensemble de descripteurs de leur contenu, dans le but de mesurer la ressemblance avec les descripteurs correspondant à la requête.

# Contexte

## Challenge ...



what the computer sees

what the human eye sees

# Indexer ?

- Indexer = extraire une information synthétique des images (documents multimédia) afin de faciliter l'accès à leur contenu
  - information = élément susceptible d'être codé pour être conservé, traité, communiqué
  - index = clé d'accès à l'information contenue dans l'image
- Pourquoi indexer ?
  - indexer pour retrouver, trier : conservation d'un patrimoine (culturel, scientifique, ...)
  - indexer pour connaître : valorisation en facilitant l'accès et l'exploration
  - indexer pour naviguer : exploitation commerciale (photos numériques, TV numérique, ...)

Le sujet de ce cours est la recherche automatique de documents visuels (images, séquences video, documents texte), dans des bases de données multimédias de grande taille, à partir de requêtes relatives au contenu de ces documents.

# Indexer ?

- Indexer = extraire une information synthétique des images (documents multimédia) afin de faciliter l'accès à leur contenu
  - information = élément susceptible d'être codé pour être conservé, traité, communiqué
  - index = clé d'accès à l'information contenue dans l'image
- Pourquoi indexer ?
  - indexer pour retrouver, trier : conservation d'un patrimoine (culturel, scientifique, ...)
  - indexer pour connaître : valorisation en facilitant l'accès et l'exploration
  - indexer pour naviguer : exploitation commerciale (photos numériques, TV numérique, ...)

Le sujet de ce cours est la recherche automatique de documents visuels (images, séquences video, documents texte), dans des bases de données multimédias de grande taille, à partir de requêtes relatives au contenu de ces documents.

# Indexer ?

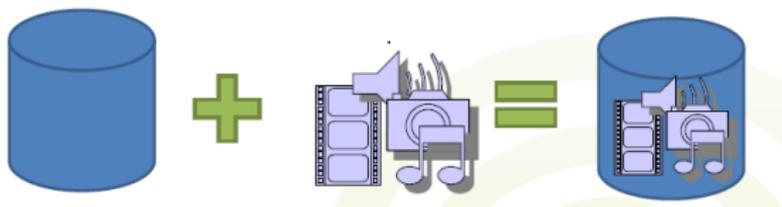
- Indexer = extraire une information synthétique des images (documents multimédia) afin de faciliter l'accès à leur contenu
  - information = élément susceptible d'être codé pour être conservé, traité, communiqué
  - index = clé d'accès à l'information contenue dans l'image
- Pourquoi indexer ?
  - indexer pour retrouver, trier : conservation d'un patrimoine (culturel, scientifique, ...)
  - indexer pour connaître : valorisation en facilitant l'accès et l'exploration
  - indexer pour naviguer : exploitation commerciale (photos numériques, TV numérique, ...)

Le sujet de ce cours est la recherche automatique de documents visuels (images, séquences video, documents texte), dans des bases de données multimédias de grande taille, à partir de requêtes relatives au contenu de ces documents.

# Définition

Quelles sont les bases de données multimédia (MMDB)?

- **Base de données (Databases) + Multimédia = BDMM**



- Une base de **données multimédia** est un type de base de données consacré au stockage, à l'organisation et à l'interrogation de données multimédia
- Ces données peuvent être de différents types, en particulier :
  - Texte, Son, Image et Vidéo
  - Qui peuvent être combinés (exemple : film)

# BDMM

## Pourquoi les bases de données mutlimédia ?

### Quelques chiffres

- Croissance très importante, en raison de l'accumulation des contenus numériques (image, vidéo, ...) auto-produits par le grand public, par exemple :
  - Facebook : plus de 100 millions d'images/jour
  - Youtube : plus 24h de vidéo/minute
  - Films : <http://www.imdb.org> recense plus de 400 000 films
- Internet : nécessite Plus 5-10 To de capacité de stockage
  - Sept 2011 : 1 milliards de recherches/jour
- Fonds de document télévisuel de la SNRT (société nationale de radiodiffusion et de télévision)
  - Pour une seule chaîne TV : plus 8000h de vidéo /an
  - Pour plus 20 ans d'émission (plusieurs chaînes) : 1 million d'heures (vidéos, images, ...)

# BDMM

## Pourquoi les bases de données mutlimédia ?

### Quelques chiffres

- Croissance très importante, en raison de l'accumulation des contenus numériques (image, vidéo, ...) auto-produits par le grand public, par exemple :
  - Facebook : plus de 100 millions d'images/jour
  - Youtube : plus 24h de vidéo/minute
  - Films : <http://www.imdb.org> recense plus de 400 000 films
- Internet : nécessite Plus 5-10 To de capacité de stockage
  - Sept 2011 : 1 milliards de recherches/jour
- Fonds de document télévisuel de la SNRT (société nationale de radiodiffusion et de télévision)
  - Pour une seule chaîne TV : plus 8000h de vidéo /an
  - Pour plus 20 ans d'émission (plusieurs chaînes) : 1 million d'heures (vidéos, images, ...)

# Difficultés

- Très grands volumes de données multimédia à exploiter : Océan d'images, vidéos et son
  - Problèmes de gestion de bases de données
- "Fossé sémantique" : comment accéder à/coder une information visuelle ?, à la différence de données textuelles, le contenu sémantique n'est jamais explicite
- Les requêtes sont difficiles à exprimer, donc en général ambiguës, incomplètes



# Historique

- 1980 : Annotation textuelle des images
- 1990 : Indexation par la couleur, la forme et la texture
- 1993 : Requête en utilisant la similarité des images
- 1997 : Requête par images exemplaires et retour de la pertinence (relevance feedback)
- 2000 : Apprentissage sémantique et adaptation à l'utilisateur
- Enjeux actuels : Deep learning in computer vision, Convolutional Neural Networks for Visual Recognition

# Quelques Système de Recherche Multimédia

- Quelques Système de Recherche Multimédia
  - Systèmes industriels
    - QBIC (IBM, depuis 1995) : <http://www.qbic.almaden.ibm.com>
    - ImageFinder (Attrasoft) : <http://www.attrasoft.com>
    - Virage (Virage Technologies, 1996) : <http://www.virage.com>
  - Systèmes académiques
    - IMEDIA (INRIA) : <http://www-rocq.inria.fr/imedia/>
    - SAPIR (projet Européen) : <http://milos.isti.cnr.it:8080/milos/album/>
    - Tiltomo : <http://www.tiltomo.com>
- Quelques SGBD standard pour BDMM
  - Jasmine est une base de données orientée objet, livré avec quatre classes multimédias: (photo, image, vidéo et audio)
  - ORACLE (ORACLE Multimedia, le package ORDSYS, les classes ORDVideo, ORDAudio et ORDImage)
  - IBM DB2 Extenders

# Quelques Système de Recherche Multimédia

- Quelques Système de Recherche Multimédia
  - Systèmes industriels
    - QBIC (IBM, depuis 1995) : <http://www.qbic.almaden.ibm.com>
    - ImageFinder (Attrasoft) : <http://www.attrasoft.com>
    - Virage (Virage Technologies, 1996) : <http://www.virage.com>
  - Systèmes académiques
    - IMEDIA (INRIA) : <http://www-rocq.inria.fr/imedia/>
    - SAPIR (projet Européen) : <http://milos.isti.cnr.it:8080/milos/album/>
    - Tiltomo : <http://www.tiltomo.com>
- Quelques SGBD standard pour BDMM
  - Jasmine est une base de données orientée objet, livré avec quatre classes multimédias: (photo, image, vidéo et audio)
  - ORACLE (ORACLE Multimedia, le package ORDSYS, les classes ORDVideo, ORDAudio et ORDImage)
  - IBM DB2 Extenders

# Quelques Système de Recherche Multimédia

- Quelques Système de Recherche Multimédia
  - Systèmes industriels
    - QBIC (IBM, depuis 1995) : <http://www.qbic.almaden.ibm.com>
    - ImageFinder (Attrasoft) : <http://www.attrasoft.com>
    - Virage (Virage Technologies, 1996) : <http://www.virage.com>
  - Systèmes académiques
    - IMEDIA (INRIA) : <http://www-rocq.inria.fr/imedia/>
    - SAPIR (projet Européen) : <http://milos.isti.cnr.it:8080/milos/album/>
    - Tiltomo : <http://www.tiltomo.com>
- Quelques SGBD standard pour BDMM
  - Jasmine est une base de données orientée objet, livré avec quatre classes multimédias: (photo, image, vidéo et audio)
  - ORACLE (ORACLE Multimedia, le package ORDSYS, les classes ORDVideo, ORDAudio et ORDImage)
  - IBM DB2 Extenders

# Deep Learning Frameworks

## Caffe

No need to write code!

1. Convert data (run a script)
2. Define net (edit prototxt)
3. Define solver (edit prototxt)
4. Train (with pretrained weights)

## TensorFlow

```
import tensorflow as tf
import numpy as np

# Model parameters
N, H, K, C = 10, 100, 100, 30

# Input data
x = tf.placeholder(tf.float32, shape=[None, N])
y = tf.placeholder(tf.float32, shape=[None, C])

# Weights and biases
w1 = tf.Variable(x * 0.1, name='w1')
w2 = tf.Variable(x * 0.1, name='w2')

# Bias
b1 = tf.Variable(0.1, name='b1')
b2 = tf.Variable(0.1, name='b2')

# Forward pass
z1 = tf.matmul(x, w1) + b1
z2 = tf.matmul(z1, w2) + b2

# Softmax
softmax = tf.nn.softmax(z2)

# Loss and accuracy
loss = tf.nn.softmax_cross_entropy_with_logits(logits=z2, labels=y)
accuracy = tf.nn.accuracy(logits=z2, labels=y)
```

## Torch

```
require 'torch'
require 'nn'

-- Model parameters
N, H, K, C = 10, 100, 100, 30

-- Input data
x = torch.rand(N, N)
y = torch.rand(N, C)

-- Weights and biases
w1 = torch.rand(N, N)
w2 = torch.rand(N, N)
b1 = torch.rand(N)
b2 = torch.rand(N)

-- Forward pass
z1 = torch.add(torch.matmul(x, w1), b1)
z2 = torch.add(torch.matmul(z1, w2), b2)

-- Softmax
softmax = torch.nn.Softmax:apply(z2)

-- Loss and accuracy
loss = torch.nn.CrossEntropyLoss:apply(z2, y)
accuracy = torch.nn.AverageValue:apply(softmax)
```

## Theano

```
import theano
import theano.tensor as T

# Model parameters
N, H, K, C = 10, 100, 100, 30

# Input data
x = T.matrix('x')
y = T.matrix('y', dtype='int32')
w1 = T.matrix('w1')
w2 = T.matrix('w2')

# Forward pass: Compute scores
s = x.dot(w1)
a, r = T.nnet.relu(s)
scores = a.dot(w2)

# Forward pass: compute softmax loss
prob = T.nnet.softmax(scores)
loss = T.nnet.categorical_crossentropy(prob, y).sum()

# Backward pass: compute gradients
d_w2 = T.grad(loss, [w2])

# Theano function
f = theano.function(
    inputs=[x, y, w1, w2],
    outputs=[prob, loss, d_w2])
```

## Lasagne

```
import theano
import theano.tensor as T

# Model parameters
N, H, K, C = 10, 100, 100, 30

# Input data
x = T.matrix('x')
y = T.matrix('y', dtype='int32')

# Weights and biases
w1 = T.matrix('w1')
w2 = T.matrix('w2')
b1 = T.matrix('b1')
b2 = T.matrix('b2')

# Forward pass
z1 = T.add(T.dot(x, w1), b1)
z2 = T.add(T.dot(z1, w2), b2)

# Softmax
softmax = T.nnet.softmax(z2)

# Loss and accuracy
loss = T.nnet.categorical_crossentropy(softmax, y).sum()
accuracy = T.nnet.accuracy(softmax, y)
```

## Keras

```
from keras.models import Sequential
from keras.layers import Dense, Activation
from keras.optimizers import SGD
from keras.metrics import accuracy

N, H, K, C = 1000, 100, 10

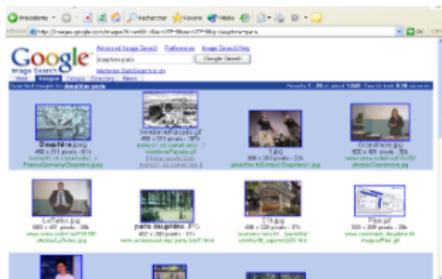
model = Sequential()
model.add(Dense(input_dim=N, output_dim=H))
model.add(Activation('relu'))
model.add(Dense(input_dim=H, output_dim=C))
model.add(Activation('softmax'))

log = model.compile(optimizer=SGD, loss='categorical_crossentropy', metrics=[accuracy])

N, H, K, C = 1000, 100, 10
X = np.random.rand(N, N)
Y = np.random.rand(N, C)
model.fit(X, Y, nb_epochs=10, batch_size=10, verbose=0)
```

# Domaines d'application

- Audiovisuel
  - Exemples : détection de copies (droits), retrouver un plan spécifique d'un programme, annotation automatique de vidéos
- Sécurité
  - Exemples : biométrie (empreintes), vidéosurveillance
- Internet
  - Exemples : commerce électronique
- ..., Bases d'images médicales



# Domaines d'application

## Exemple d'application : Sécurité

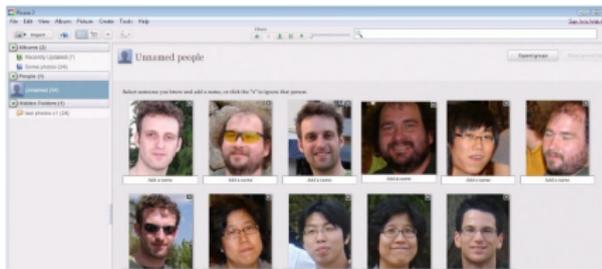
- Les caméras sont partout : à London, "il y a au moins 500 000 caméras dans la ville, et une étude a montré que dans un seul jour, une personne pourrait atteindre d'être filmé 300 fois "



# Domaines d'application

## Exemple d'application : Sécurité

- Requêtes possibles
  - Exemple de requête par mots-clés: policier veut examiner des photos de Personne X
  - Requête: "récupérer toutes les images de la base d'images dans lesquelles Personne X apparaît"
  - Requête d'image par exemple: l'agent de police a une photo et veut trouver l'identité de la personne sur la photo



# Comment indexer/interroger des documents multimédia ?

Indexation manuelle (par texte : annotation) ou automatique (par contenu) ?

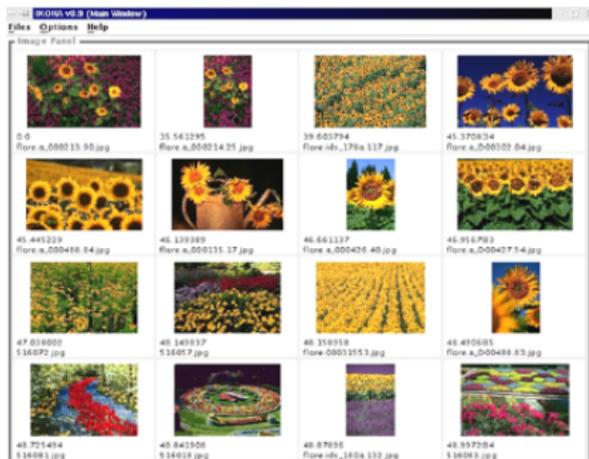
- Requête de BD Multimédia : requête peut prendre des formes très différentes, elle peut être
  - Conceptuelle (ex : mot), indexation par texte (annotation)
  - Instancielle (ex : une autre image, vidéo), exp : récupérer tous les images similaires à une image requête
  - ...



# Indexation manuelle ou automatique ?

Type de requête ==> Type d'indexation

- Recherche par contenu
  - Qu'est-ce que ces images ont exactement en commun?



- Ou recherche par texte
  - Récupérer toutes les images montrant : "tournesol"

# Indexation manuelle

## ● Annotation

- l'approche dite classique consiste à indiquer des mots-clés attachés au document (ou indexés) décrivant, dans un vocabulaire restreint, les caractéristiques principales et bien identifiables des documents stockés.
- Par quoi indexer : mots-clés, métadonnées, ...



Mots-clés : tournesol, soleil

## ● Inconvénients

- Même image peut avoir plusieurs annotations différentes
- Ambiguïté de l'annotation
- Dépendance du contexte
- Le coût d'annotation manuel est très important (10 fois la durée de document)

## ● Approche la plus ancienne et la plus répandue

# Indexation manuelle

## ● Annotation

- l'approche dite classique consiste à indiquer des mots-clés attachés au document (ou indexés) décrivant, dans un vocabulaire restreint, les caractéristiques principales et bien identifiables des documents stockés.
- Par quoi indexer : mots-clés, métadonnées, ...



Mots-clés : tournesol, soleil

## ● Inconvénients

- Même image peut avoir plusieurs annotations différentes
- Ambiguïté de l'annotation
- Dépendance du contexte
- Le coût d'annotation manuel est très important (10 fois la durée de document)

## ● Approche la plus ancienne et la plus répandue

# Indexation manuelle

Dépendance du contexte



# Indexation automatique

- Indexation par contenu
  - L'algorithme d'indexation attache des données de bas niveau sémantique, relatifs aux contenus géométrique, spectral, de l'image, à un niveau local ou global.



- Recherche par contenu (CBIR : Content-Based Information Retrieval)
  - Les requêtes se font en général par l'exemple, ou par modèle.

Extraire automatiquement d'une image (ou document multimédia) des descripteurs significatifs et compacts, qui seront utilisés pour la recherche ou la structuration

# Indexation automatique

- Indexation par contenu
  - L'algorithme d'indexation attache des données de bas niveau sémantique, relatifs aux contenus géométrique, spectral, de l'image, à un niveau local ou global.



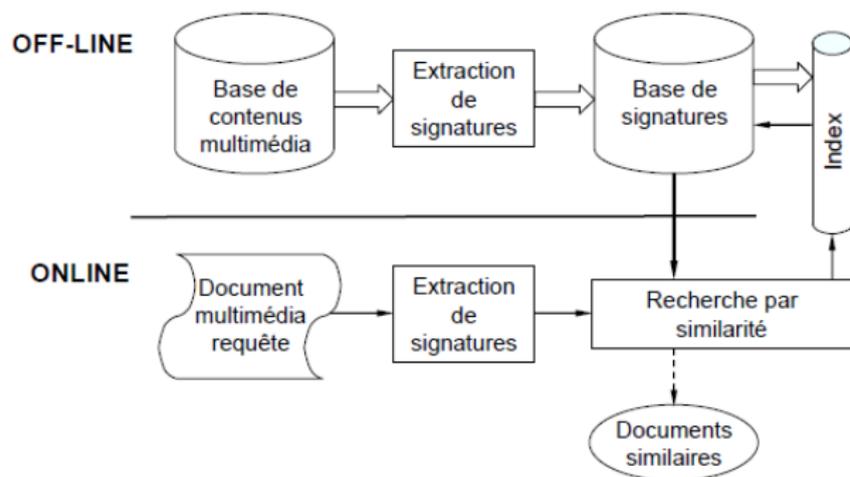
- Recherche par contenu (CBIR : Content-Based Information Retrieval)
  - Les requêtes se font en général par l'exemple, ou par modèle.

Extraire automatiquement d'une image (ou document multimédia) des descripteurs significatifs et compacts, qui seront utilisés pour la recherche ou la structuration

# Indexation automatique

## BDMM : Architecture générale

- Off-line : production d'indexes issus de l'analyse du contenu des images (extraction de caractéristiques pertinentes, organisation...)



- On-line : gestion des requêtes des utilisateurs

# Définitions

## Indice, Descripteur, Signature

- Indices visuels : caractéristiques de l'image, au sens de perception humaine, que l'on cherche à utiliser pour la tâche considérée
  - Principaux indices visuels : couleur, forme, texture, régions, mouvement (essentiellement pour la vidéo) (Ici, focus sur l'image, mais s'applique à la vidéo)
- Descripteur d'image : méthode d'extraction du contenu visuel de l'image
  - Exemple : histogramme couleur
- Signature d'image (caractéristiques) : vecteur numérique représentant le contenu visuel de l'image
  - Exemple : 1 vecteur de dimension 216 pour l'histogramme couleur
- Espace de description (de représentation) des images
  - 1 image = 1 ou plusieurs points dans un espace multidimensionnel
- Espace de recherche dans la base d'images
  - Structuration de l'espace de description pour une recherche efficace (index)

# Mesure de similarité

## Distance

- En mathématiques, on appelle distance sur un ensemble  $E$  une application  $d$  définie sur le produit  $E^2 = E \times E$  et à valeurs dans l'ensemble  $R^+$  des réels positifs

$$d : E \times E \rightarrow R^+$$

... vérifiant les propriétés suivantes :

Nom	Propriété
Symétrie	$\forall (a, b) \in E^2, d(a, b) = d(b, a)$
Séparation	$\forall (a, b) \in E^2, d(a, b) = 0 \Leftrightarrow a = b$
Inégalité triangulaire	$\forall (a, b, c) \in E^3, d(a, c) \leq d(a, b) + d(b, c)$

- Un ensemble muni d'une distance est un espace métrique

# Mesure de similarité

## Distance

- Dans  $R^n$ , on peut définir de plusieurs manières la distance entre deux points
- Soient deux points de  $E$ ,  $(x_1, x_2, \dots, x_n)$  et  $(y_1, y_2, \dots, y_n)$ , on exprime les différentes distances ainsi

Nom	Paramètre	Fonction
distance de Manhattan	1-distance	$\sum_{i=1}^n  x_i - y_i $
distance euclidienne	2-distance	$\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
distance de Minkowski	p-distance	$\sqrt[p]{\sum_{i=1}^n  x_i - y_i ^p}$
distance de Tchebychev	$\infty$ -distance	$\lim_{p \rightarrow \infty} \sqrt[p]{\sum_{i=1}^n  x_i - y_i ^p}$ $= \sup_{1 < i < n}  x_i - y_i $

# OpenCV

## Introduction

- Bibliothèque de traitement d'images et de vision par ordinateur en langage C/C++, python, java
- Projet initié par Intel (en 1999, actuellement 3.)
- Multi-plateformes : disponible pour Linux, Windows et Mac OS
- Elle est Open Source
- Fortement orientée temps réel
- Optimisée pour les processeurs multi-coeurs
- Que contient-elle ?
  - Des fonctionnalités (structures et routines,  $\geq 2500$  algos) de base et avancées
  - pour bâtir rapidement des applications autour de la vision par ordinateur.
- Téléchargeable sur <http://sourceforge.net/projects/opencvlibrary/>

# OpenCV

## Introduction

OpenCV regroupe plusieurs modules :

- **Core** : contient les fonctionnalités de base, notamment l'accès aux pixels, le changement de luminosité, de contraste, le changement d'espace couleur, la possibilité de dessiner sur les images, etc.
- **Imgproc** :
  - permet d'appliquer différents filtres (moyen, gaussien, médian)
  - permet d'appliquer différentes opérations morphologiques (dilatation, érosion, ouverture, fermeture, gradient morphologique etc)
  - contient différents algorithmes de seuillage et de détection de contours (Sobel, opérateur de Laplace, Canny Edge Detector), mais aussi des détecteur de droites ou d'ellipse (Hough Line/Cercle Transform).
- **Highgui** : permet l'ajout de composants graphiques de base, mais aussi avancés.
- **Calib3D** : permet la calibration des caméras et la reconstruction 3D

# OpenCV

## Introduction

- **Feature2D** : contient des descripteurs 2D souvent utilisés basés sur les couleurs, la forme, la texture, les points d'intérêt ... et des algorithmes de mises en correspondance.
- **Video** : contient les fonctionnalités de base de traitement de vidéos (algorithmes de détection de mouvement, de suivi, d'extraction de plan principal etc. ...).
- **Objdetect** : contient des algorithmes de détection d'objet, notamment la détection de visages ...
- **ML** : contient les algorithmes d'apprentissage et de classification.
- **GPU** : contient les algorithmes permettant l'utilisation de la carte graphique afin d'accélérer le temps d'exécution grâce au parallélisme des GPU.

# OpenCV

## Classes de bases

- La classe **Point\_**
  - Structure de données générique pour représenter des points dans espace de dimension 2.

```
Template <typename _Tp> class Point_ {  
    public: Point_();  
    Point_(_Tp _x, _Tp _y);  
    Point_(const Point_& pt); ... _Tp x, y;  
};
```

- Pour les types courants : points à coordonnées entières, flottantes, simple ou double précision, on dispose des alias suivants:

```
typedef Point_<int> Point2i;  
typedef Point2i Point;  
typedef Point_<float> Point2f;  
typedef Point_<double> Point2d ;  
  
// Exemple d'utilisation :  
Point2f a(0.3f, 0.f), b(0.f, 0.4f);  
Point pt = (a + b)*10.f;  
cout << pt.x << ", " << pt.y << endl;
```

# OpenCV

## Classes de bases

- Dimension **Size\_**
  - Représente la taille d'un objet rectangulaire a 2 dimensions. Comme pour les points il s'agit d'une structure générique.

```
Template <typename _Tp> class Size_  
{  
public:  
    Size_();  
    Size_(Tp _width, _Tp _height);  
    Size_(const Size_& sz);  
    _Tp area() const;  
    ...  
    _Tp width, height;  
};
```

- Et on dispose des alias suivant :

```
//Et on dispose des alias suivant :  
typedef Size_<int> Size2i;  
typedef Size2i Size;  
typedef Size_<float> Size2f;
```

# OpenCV

## Classes de bases

- Vecteur **Vec**

- Représente un vecteur générique de faible dimension ( $\leq 10$ ).

```
Template <typename _Tp, int cn> class Vec
{
public:
// default constructor: all elements are set to 0
Vec();

// constructors taking up to 10 first elements as parameters
Vec(_Tp v0);
Vec(_Tp v0, _Tp v1);
...
Vec(_Tp v0, _Tp v1, _Tp v2, _Tp v3, _Tp v4, _Tp v5, _Tp v6,
_Tp v7, _Tp v8, _Tp v9);
Vec(const Vec<_Tp, cn>& v);

// constructs vector with all the components set to alpha.
static Vec all(_Tp alpha);
...
//data
_Tp val[cn];
};
```

# OpenCV

## Classes de bases

- Vecteur **Vec**
  - Et on dispose des alias suivant :

```
typedef Vec<uchar, 2> Vec2b;  
typedef Vec<uchar, 3> Vec3b;  
typedef Vec<uchar, 4> Vec4b;  
...  
typedef Vec<double, 6> Vec6d;
```

# OpenCV

## Classes de bases

- La classe **MAT**

- La classe Mat (pour matrice) permet de stocker l'image sous forme matricielle.
- Chaque objet Mat possède deux parties :
  - L'entête
  - Les données
- Allocation par constructeur : La classe Mat propose un nombre important de constructeur, viola ceux qui vous seront le plus utile :

```
Mat::Mat(int rows, int cols, int type)

Mat::Mat(Size size, int type)

//Exemple :
// allocate a matrix of 1024 rows by 768 columns with 32 bits
floating precision and 3 channels
Mat im1(1024,768,CV_32FC3);
Size s(768,1024);
// allocate a matrix of 1024 rows by 768 columns with unsigned
8 bits integer precision and 1 channel
Mat im2(s,CV_8UC1);
```

# OpenCV

## Classes de bases

- La classe **MAT**

- Types : représenté par  $CV_ < Datatype > C < Channels >$ , tels que **CV\_8UC1**, **CV\_32UC3**

OpenCV Tag	Representation	OpenCV Value
CV_8U	8 bit unsigned integer	0
CV_8S	8 bit signed integer	1
CV_16U	16 bit unsigned integer	2
CV_16S	16 bit signed integer	3
CV_32S	32 bit signed integer	4
CV_32F	32 bit floating point number	5
CV_64F	64 bit floating point number	6