Introduction à la Vision par Odinateur

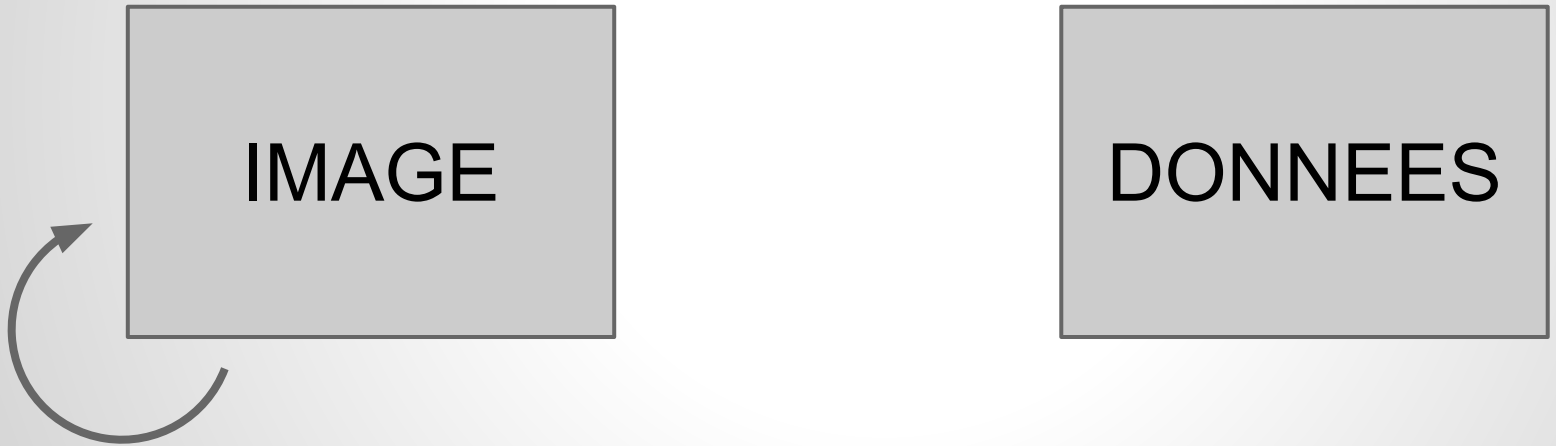
via OpenCV

Vision par Ordinateur ???

IMAGE

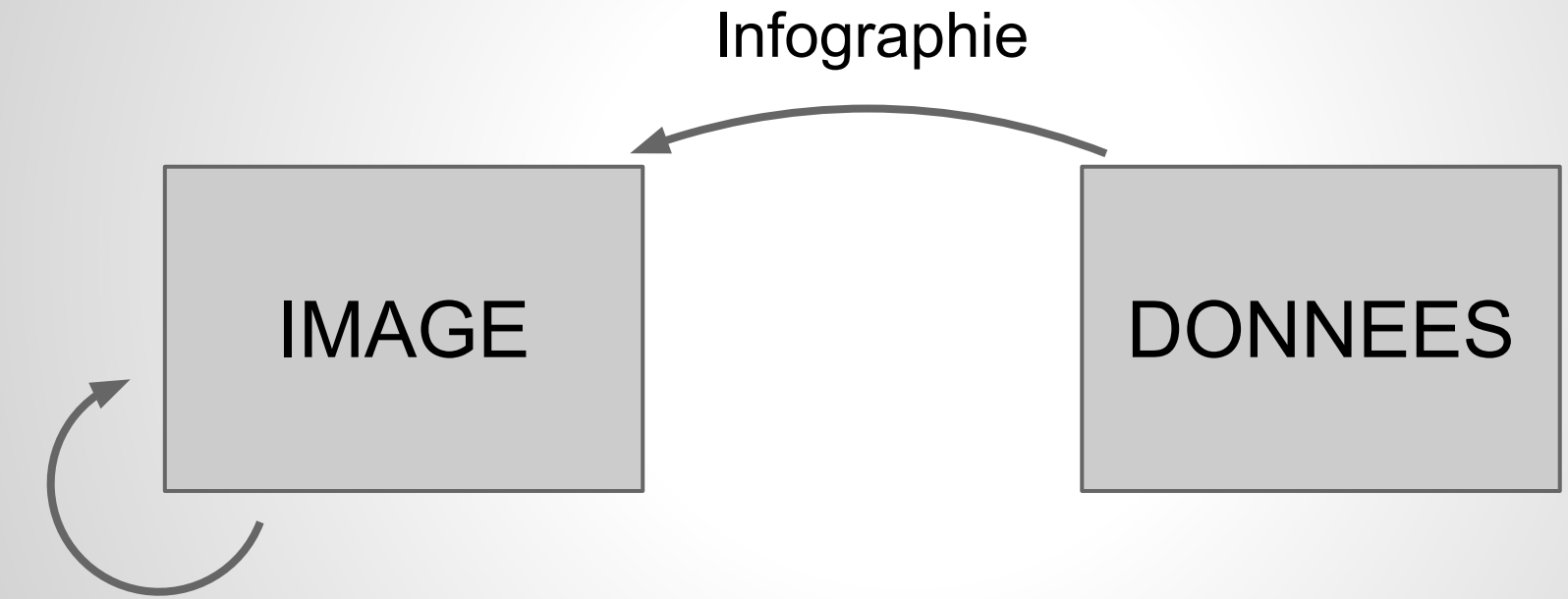
DONNEES

Vision par Ordinateur ???

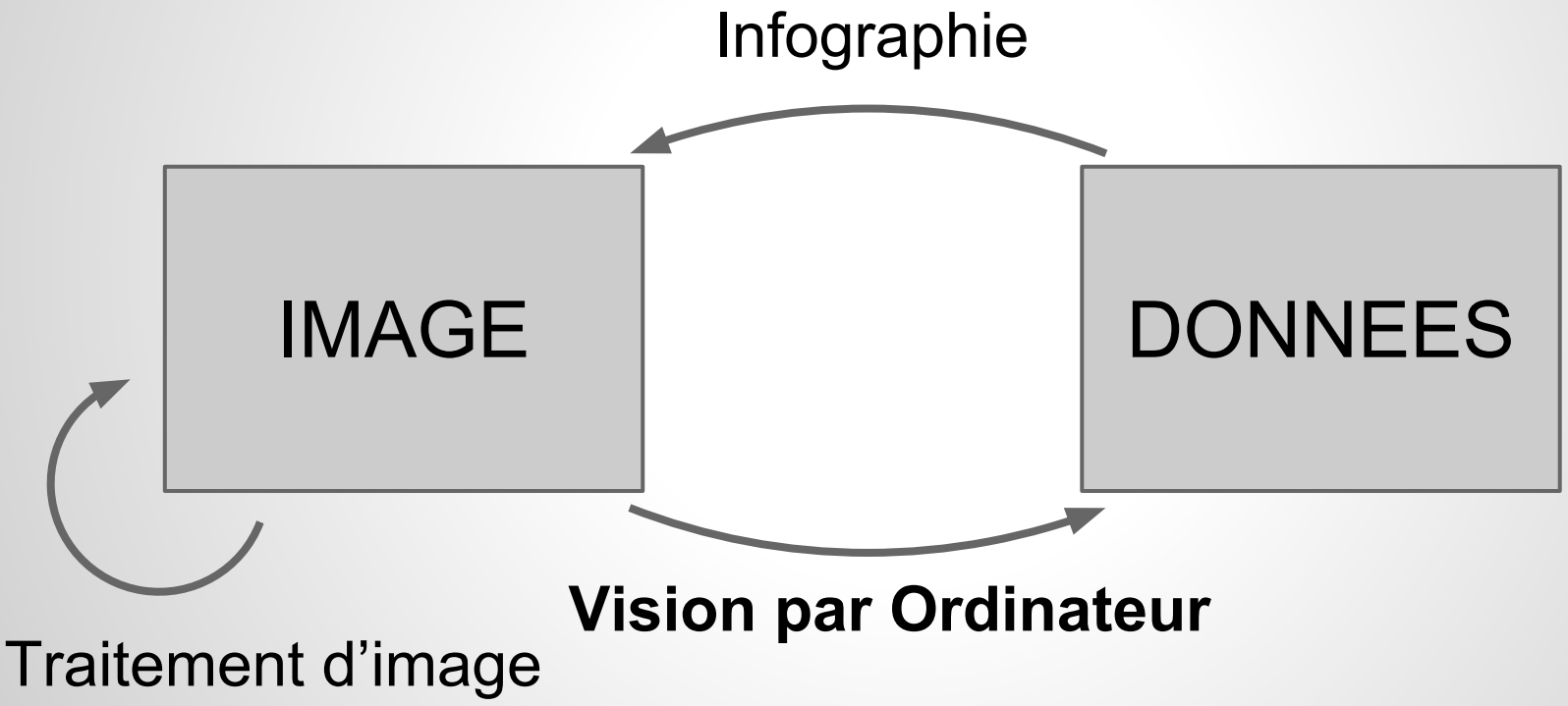


Traitement d’image

Vision par Ordinateur ???



Vision par Ordinateur ???



Applications

- Médical
- Industrie
- Sécurité
- Robotique
- Divertissement
- Information
- ….

OpenCV ???

OpenCV est une librairie :

- de vision par ordinateur
- gratuite
- très utilisée dans le milieu
- écrite en C++
- avec des portages Python et Java
- compatible Windows, Linux, iOs, Android

Installer OpenCV

Tout est sur le site :

Windows:

http://docs.opencv.org/doc/tutorials/introduction/windows_install/windows_install.html

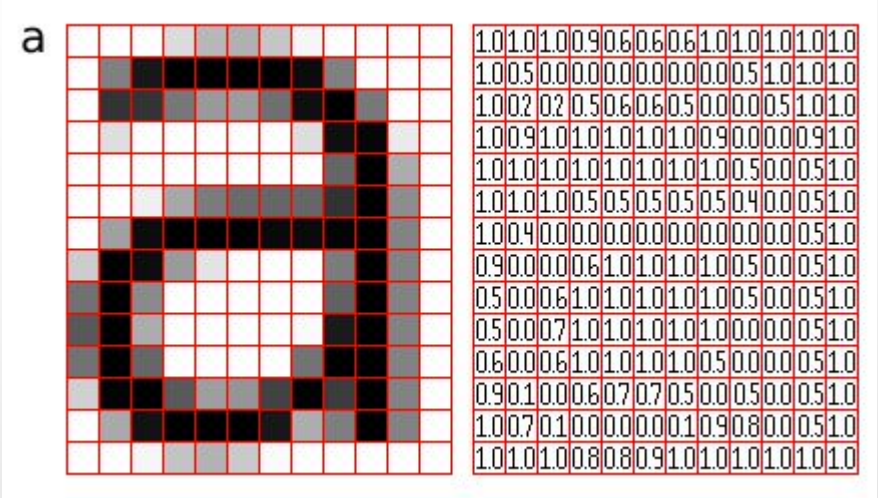
Linux :

http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html

Représentation d'une image

OpenCV manipule des images **matricielles :**

l'image est divisée en cases (pixels), contenant une ou plusieurs valeurs définissant sa couleur.



Représentation d’une image

Une image matricielle est caractérisée par un certain nombre de propriétés :

- ses dimensions
- son type de données
- son nombre de canaux
- le contenu de sa matrice

Dimensions et coordonnées

- Les dimensions d'une images sont le plus souvent représentées en **Largeur x Hauteur.**
- L'origine d'une image est le plus souvent son coin supérieur gauche.
- Les coordonnées vont de (0,0) à (largeur-1, hauteur-1)

Type de données

Les images stockent le plus souvent les valeurs sur un octet non signé : 256 niveaux sont suffisants pour que l'oeil ait une impression de continuité.

D'autres types de valeurs peuvent être utiles pour les manipulations intermédiaires.

Nombre de canaux

Les images sous OpenCV peuvent avoir :

- 1 seul canal : niveaux de gris/binaires
- 3 canaux : couleur -> RGB, HSV, HSL, etc…
- 4 canaux : ARGB

Les modules OpenCV

OpenCV est composé en modules, pour n’avoir à compiler/linker que ce qu’on utilise.

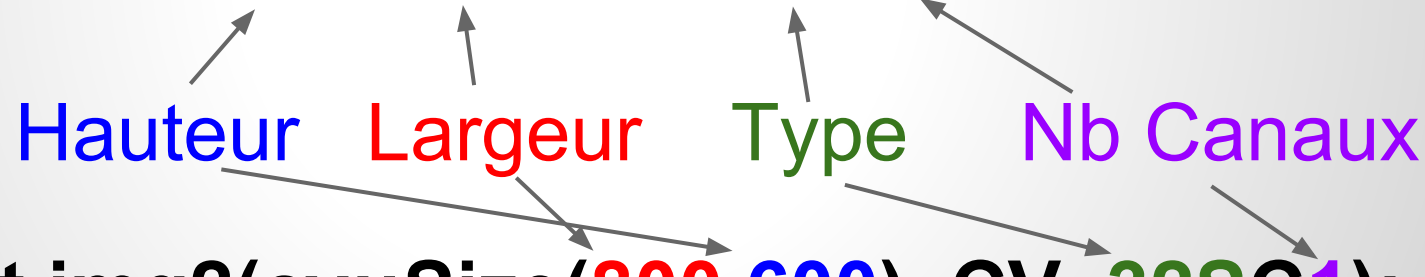
Les plus importants sont :

- core : contient les types de base
- imgproc : contient les algorithmes de base
- highgui : contient les I/O des images et vidéos

Les images dans OpenCV

OpenCV utilise le type **cv::Mat**, qui sert pour tous ses tableaux 2D, matrices ou images.

```
cv::Mat img(600, 800, CV_8UC3);
```




```
cv::Mat img2(cv::Size(800,600), CV_32SC1);
```

Charger une image

```
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/core/core.hpp>
...
cv::Mat img = cv::imread(“my_image.jpg”);
...
```

Afficher une image

```
//afficher l'image img dans la fenêtre "window"
cv::imshow("window", img);

//attendre que l'utilisateur appuie sur une touche pour continuer
cv::waitKey(0);
```

Sauvegarder une image

```
//ecrit l'image img dans le fichier "out.jpg"
cv::imwrite("out.jpg", img);
```

Représentation de la couleur

Par défaut, OpenCV représente les images couleur en BGR : Blue Green Red

Les images chargées sont codées sur 8bits non signés et en 3 canaux, le code de type est donc CV_8UC3.

Représentation de la couleur

Par défaut, OpenCV représente les images couleur en **BGR** : **B**lue **G**reen **R**ed

Les images chargées sont codées sur 8bits non signés et en 3 canaux, le code de type est donc CV_8UC3.

Conversion de format couleur

Pour changer le type de représentation, une méthode utile est :

cvtColor(src, dst, code)

Dst doit avoir le même type de données et les mêmes dimensions que src.

Conversion de format couleur

Quelques codes usuels :

CV_BGR2GRAY : converti en niveaux de gris (luminance). La sortie ne doit avoir qu'un canal.

CV_GRAY2BGR : reconverti en "couleur". Tous les canaux ont les mêmes valeurs.

CV_BGR2HSV : converti en HSV, très utile pour détecter des couleurs particulières.

Séparer les canaux

On peut séparer les canaux d'une image en plusieurs images distinctes avec :

cv::split(input, output)

input étant l'image multi canaux

et output un tableau ou un vecteur d'images, qui seront redimensionnées au besoin

Fusionner les canaux

On peut également fusionner plusieurs images distinctes de même taille et tyoe en une seule avec : **cv::merge(vec_input, output)**

ou cv::merge(tab_input, nb, output)

vec_input étant un vecteur d'images, tab_input un tableau d'images (nb sa taille), et output le résultat.

Opérations sur images

Les opérations de cv::Mat sont surchargées, on peut donc faire des choses comme :

```
cv::Mat res = mat1-mat2;
res = res*0.5;
cv::Mat matThres = res > 120;
cv::Mat diff = mat1 != mat2;
```

Opérations sur images

Toutes les autres opérations élémentaires, comme abs, min, max, etc… sont également disponibles dans le module core.

La liste est visible dans la doc de core, rubrique Operations on Arrays :

http://docs.opencv.org/modules/core/doc/operations_on_arrays.html