



# Traitemet d'Image avec OpenCV

*Aperçu sur les fonctions de la librairie*

**Séminaire Captronic 28/06/2018**

*Julien Arzi (julien.arzi@tsdconseil.fr)*

# Plan

- 1 Représentation des images
  - Espaces de couleur classiques
  - Algorithmes de détection simple
- 2 Filtrage
  - Filtrage non linéaire (médian, bilatéral)
  - Filtrage directif
  - Traitements dans le domaine fréquentiel
- 3 Reconnaissance de formes simples
  - Canny
  - Transformée de Hough
- 4 Segmentation d'image
  - Seuillage
- Ligne de partage des eaux
- Superpixels
- Applications : recherche de zones d'intérêt
- 5 Reconnaissance d'objets
  - Détection d'objets génériques
  - Détection de points d'intérêts
- 6 Traitements vidéo
  - Calcul de flux optique
- 7 Vision 3d
- 8 Autres fonctions
- 9 Formations

# TSD : Conseil en Traitement du Signal Digital

Accompagnement et formation

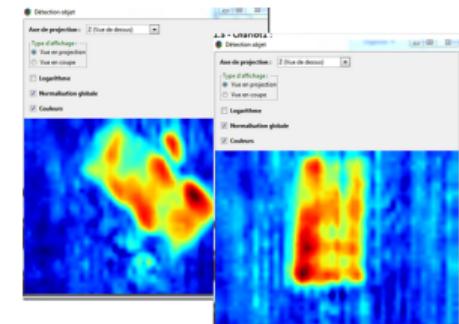
## • Traitement d'image :

- ▶ Identification à partir des empreintes digitales,
- ▶ Contrôle d'accès (codes à bulles),
- ▶ Comptage de véhicules,
- ▶ Calibration automatique de projecteur.



## • Radio logicielle :

- ▶ Modulateurs / démodulateurs embarqués sur DSP (DMR),
- ▶ SDR (*Software Defined Radio*) : traitement radio sur PC.  
Réseau SIGFOX (carte son), comms. satellite (USRP)
- ▶ Radar SAR : reconstitution 3D à partir d'un réseau d'antennes.



# OpenCV : généralités

- OpenCV = **Open Computer Vision**
- Projet initié par **INTEL** (Intel Russia / Nizhny Novgorod) en 1999
- Logiciel libre (BSD)
- Primitives très optimisées
- Cœur en **C/C++**
- Langages (Python, JAVA, etc).
- **Portabilité**

# Plan

## 1 Représentation des images

- Espaces de couleur classiques
- Algorithmes de détection simple

## 2 Filtrage

- Filtrage non linéaire (médian, bilatéral)
- Filtrage directif
- Traitements dans le domaine fréquentiel

## 3 Reconnaissance de formes simples

- Canny
- Transformée de Hough

## 4 Segmentation d'image

- Seuillage

## • Ligne de partage des eaux

## • Superpixels

## • Applications : recherche de zones d'intérêt

## 5 Reconnaissance d'objets

- Détection d'objets génériques
- Détection de points d'intérêts

## 6 Traitements vidéo

- Calcul de flux optique

## 7 Vision 3d

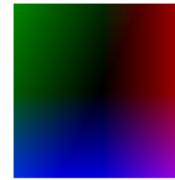
## 8 Autres fonctions

## 9 Formations

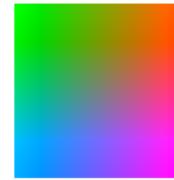
# Représentation des images

## Espaces de couleur classiques (1)

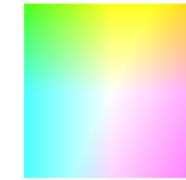
- Une image = tableau 2D (« matrice »), mono ou multicanaux, entier, flottant, etc.
- **Niveaux de gris**
- **Rouge / Vert / Bleu** (RVB / RGB) : modèle additif (format natif des écrans, capteurs CMOS / CCD, ...)
- **YC<sub>b</sub>C<sub>r</sub>/YUV** : Y = luminance ( $Y \sim 0.3R + 0.6V + 0.1B$ ),  $C_b$ ,  $C_r$  = deux canaux pour la chrominance ( $C_b \sim B - Y$ ,  $C_r \sim R - Y$ ). Exemple : formats JPEG, MJPEG, NV12, NTSC, PAL, etc.



$Y = 0$



$Y = 0.5$

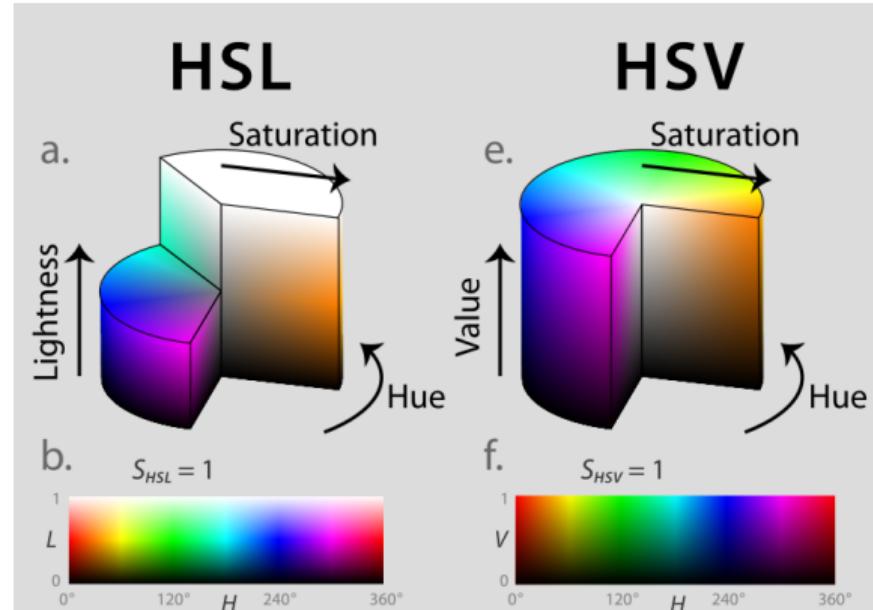


$Y = 1.0$

# Représentation des images

## Espaces de couleur classiques (2)

- **TSV / HSV** (Teinte, Saturation / chrominance, Valeur)
- **TSL / HLS** (Teinte, Saturation / chrominance, Luminance)
- Intérêt : un seul paramètre pour la teinte (couleur)
- Saturation : pureté de la teinte (saturation = 0  $\Rightarrow$  gris)

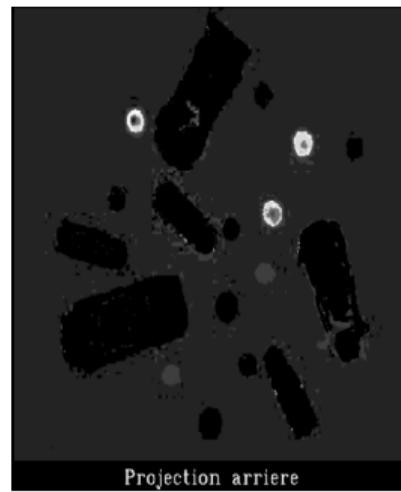


(image wikipedia)

# Algorithmes de détection simple

## Détection par colorimétrie

- Détection par **seuillage** (teinte, saturation, luminosité, etc.)
- **Projection arrière d'histogramme**
- **Distance de Mahalanobis**

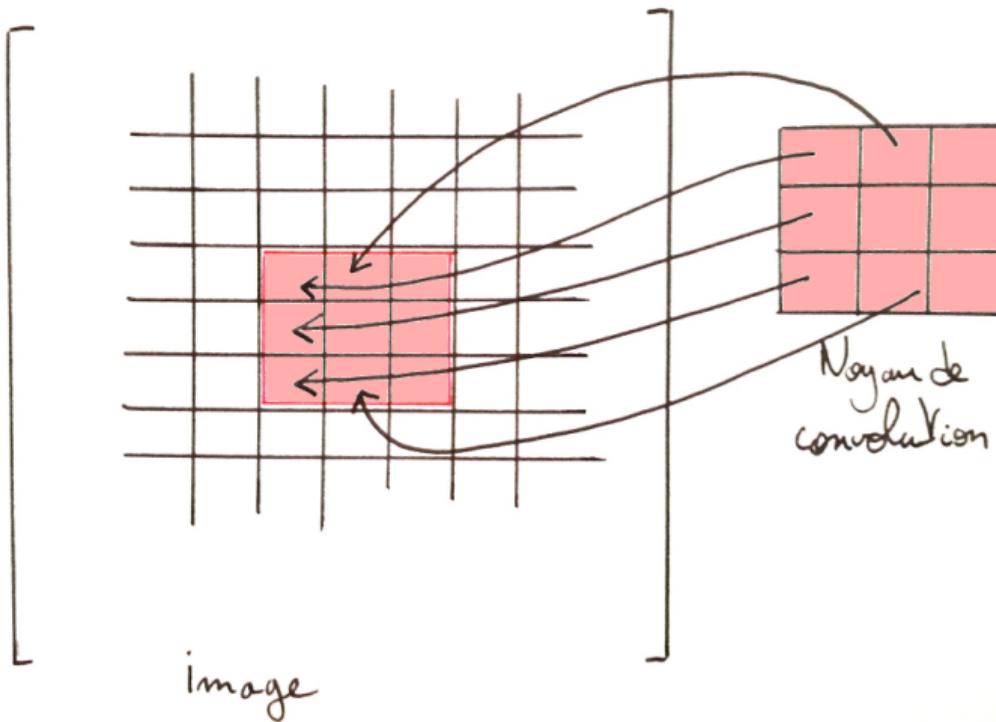


# Plan

- 1 Représentation des images
  - Espaces de couleur classiques
  - Algorithmes de détection simple
- 2 Filtrage
  - Filtrage non linéaire (médian, bilatéral)
  - Filtrage directif
  - Traitements dans le domaine fréquentiel
- 3 Reconnaissance de formes simples
  - Canny
  - Transformée de Hough
- 4 Segmentation d'image
  - Seuillage
- Ligne de partage des eaux
- Superpixels
- Applications : recherche de zones d'intérêt
- 5 Reconnaissance d'objets
  - Détection d'objets génériques
  - Détection de points d'intérêts
- 6 Traitements vidéo
  - Calcul de flux optique
- 7 Vision 3d
- 8 Autres fonctions
- 9 Formations

# Filtrage

## Filtrage linéaire FIR (produit de convolution)



# Filtrage

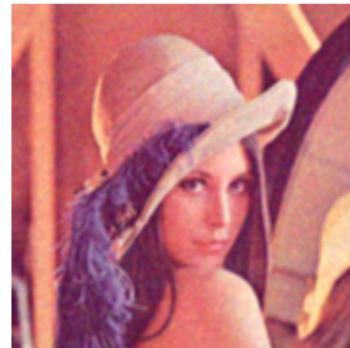
Exemples de filtres linéaires

- Filtre à **moyenne glissante** :  $K = \frac{1}{9} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$
- Filtre **gaussien** :

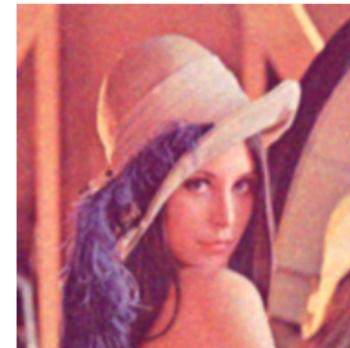
$$K(x, y) = k \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}$$



Bruitée



Glissante



Gaussien

# Filtrage

## Filtrage non linéaire

- Filtre Médian (pixels bruités isolés)
- Filtre bilatéral

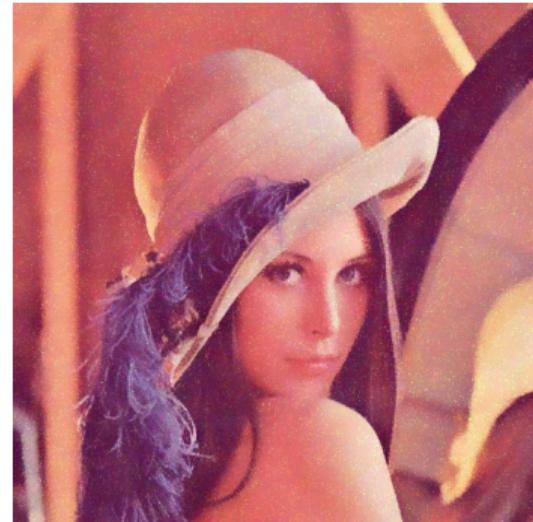


Bruitée



Bilatéral

$$\sigma_{col} = 20, \sigma_{espace} = 40$$



Bilatéral

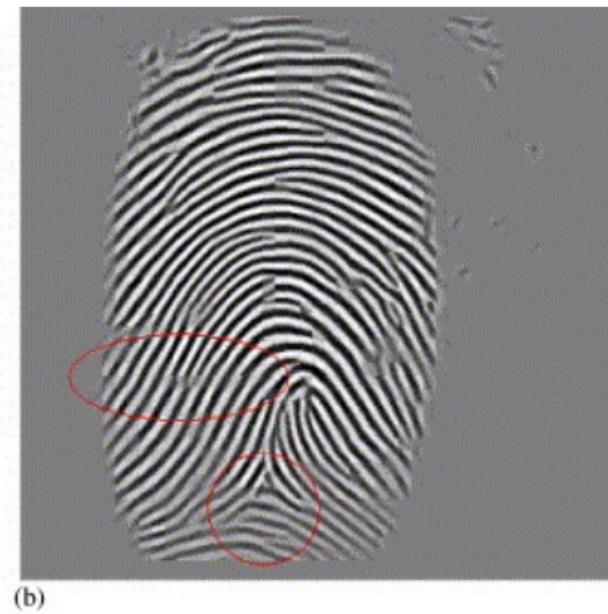
$$\sigma_{col} = 50, \sigma_{espace} = 100$$

# Filtrage

## Filtrage directif (Gabor)

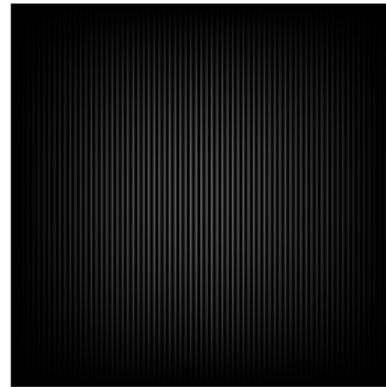
Exemples d'applications :

- Génération de **descripteurs** (reconnaissance de visages, ...)
- **Filtrage adaptatif et directionnel** par zones

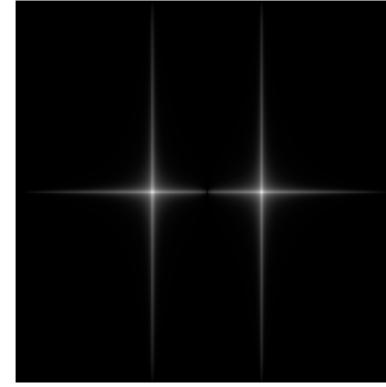


# Traitements dans le domaine fréquentiel

- **Transformée de Fourier discrète 2D**
- Entrée : image, sortie image, dans le domaine fréquentiel



Entrée



Mag TFD

- **Quelques applications :**

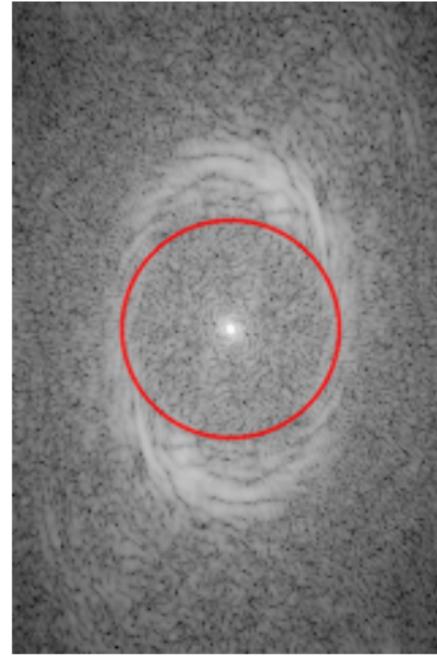
- ▶ Filtrage rapide,
- ▶ Détection de période,
- ▶ Détection de translation / rotation / échelle,
- ▶ Suppression de bruits périodiques,
- ▶ etc.

# Traitements dans le domaine fréquentiel

Exemple : détection de période



Entrée

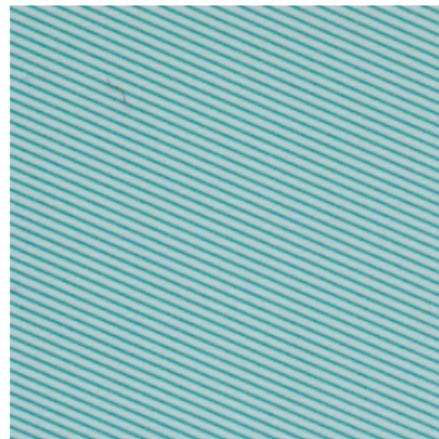


Mag TFD

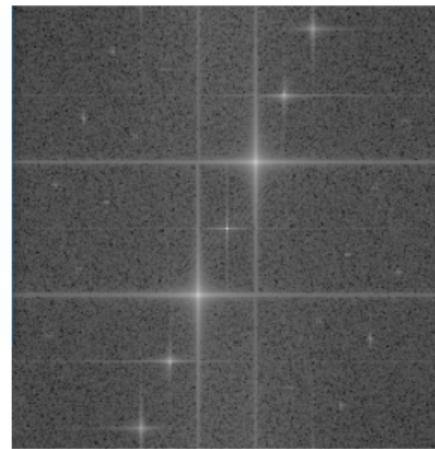
# Traitements dans le domaine fréquentiel

Exemple : soustraction spectrale

- Objectif : suppression des motifs périodiques dans une image
- Approche simple : annulation des pics dans le domaine fréquentiel



Image



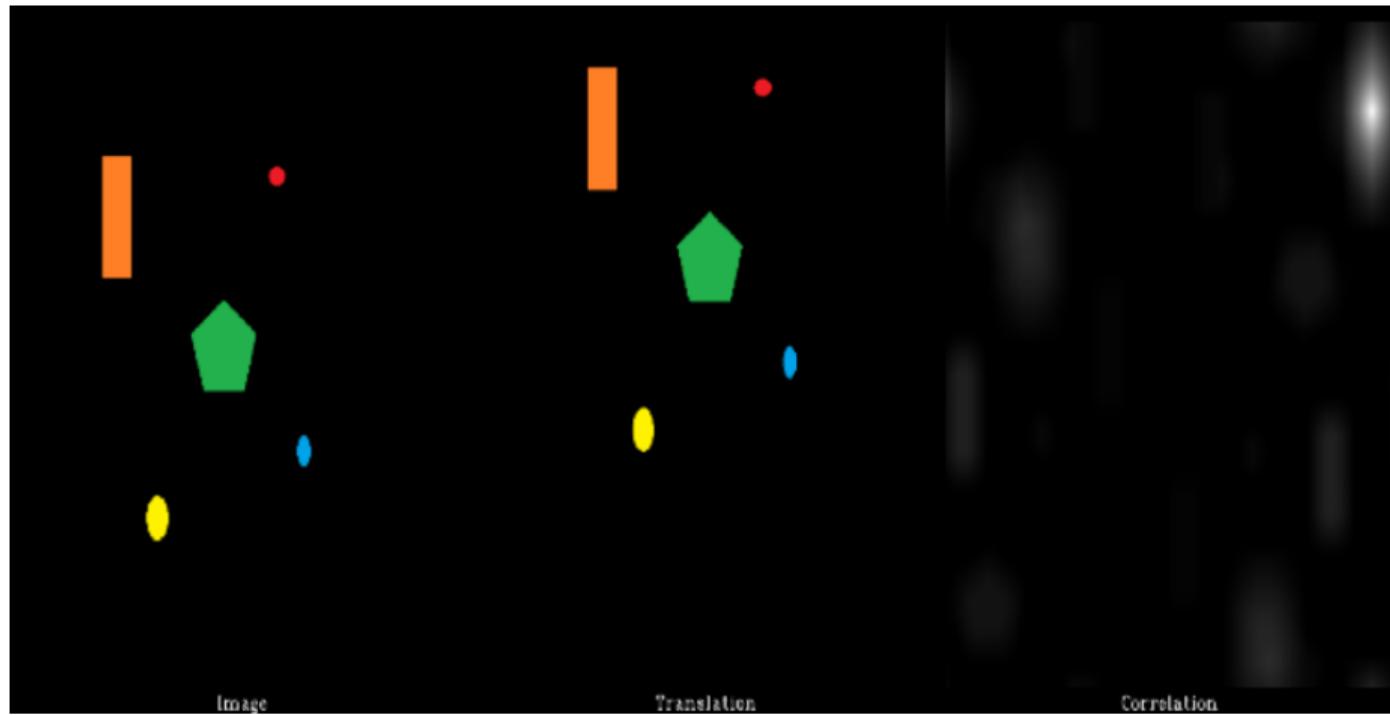
TFD



Suppression des pics

# Traitements dans le domaine fréquentiel

Exemple : détection de translation



# Plan

- 1 Représentation des images
  - Espaces de couleur classiques
  - Algorithmes de détection simple
- 2 Filtrage
  - Filtrage non linéaire (médian, bilatéral)
  - Filtrage directif
  - Traitements dans le domaine fréquentiel
- 3 Reconnaissance de formes simples
  - Canny
  - Transformée de Hough
- 4 Segmentation d'image
  - Seuillage
- Ligne de partage des eaux
- Superpixels
- Applications : recherche de zones d'intérêt
- 5 Reconnaissance d'objets
  - Détection d'objets génériques
  - Détection de points d'intérêts
- 6 Traitements vidéo
  - Calcul de flux optique
- 7 Vision 3d
- 8 Autres fonctions
- 9 Formations

# Reconnaissance de formes simples

## Calcul du gradient

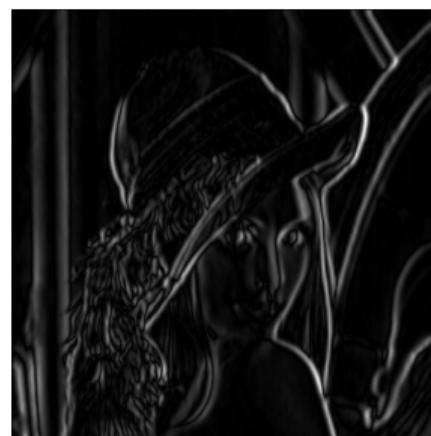
- **Objectif** : sur une image  $I$  en niveaux de gris, estimer :

$$\nabla I = \left( \frac{dI}{dx}, \frac{dI}{dy} \right)$$

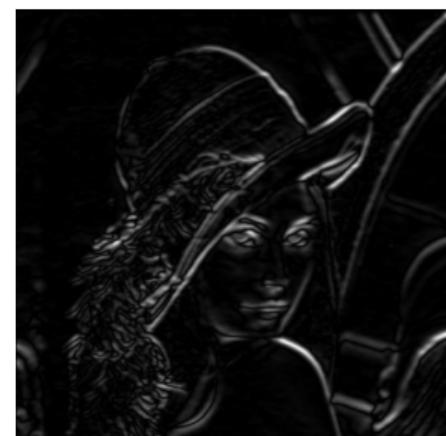
- **Applications** : extraction de contour, segmentation, incrustation d'images, flux optique, transformée de Hough, ...



Niveaux de gris



Gradient horizontal



Gradient vertical

# Reconnaissance de formes simples

## Algorithme de Canny

Étapes :

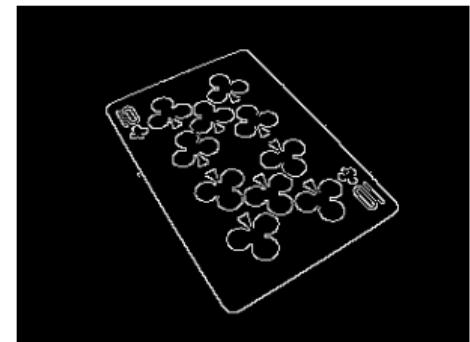
- ① Réduction du bruit (noyau gaussien)
- ② Calcul du gradient (masque de Sobel)
- ③ Suppression des non-maxima (suivant la direction du gradient)
- ④ Hystérésis (seuillage, 2 paramètres à spécifier)



Image d'entrée



Gradient

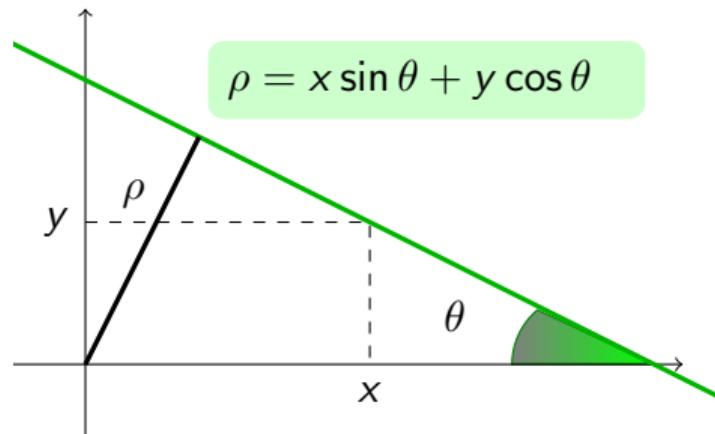


Canny

# Transformée de Hough

## Principe

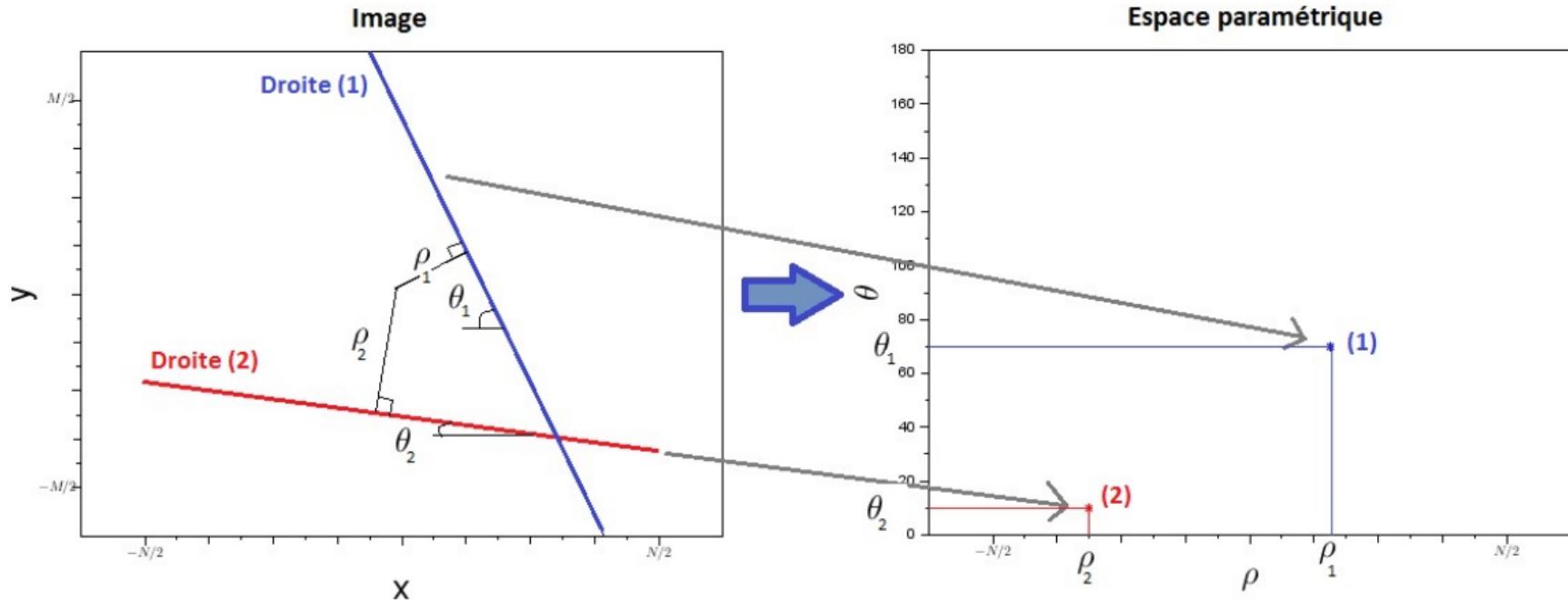
- Objectif : détection des lignes droites
- Représentation polaire d'une droite :  $(\rho, \theta)$



- Transformée de Hough = image / plan cartésien  $(x, y) \mapsto$  espace paramétrique  $(\rho, \theta)$

# Transformée de Hough

Espace de paramètre



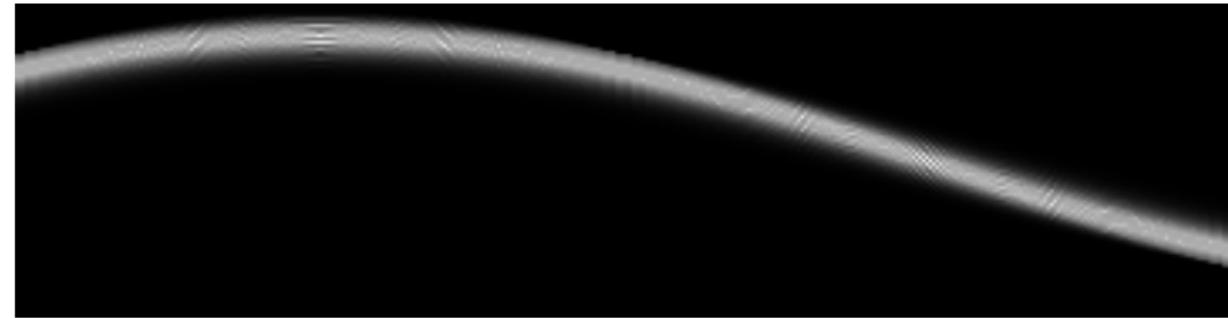
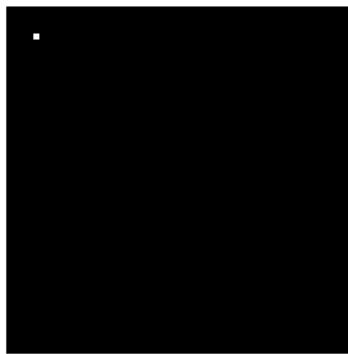
$$\rho = x \sin \theta + y \cos \theta$$

# Transformée de Hough

Algorithme basé sur la détection des contours

## Algorithme

- ① Calcul du gradient
- ② Extraction des contours
- ③ Mise à zéro d'un accumulateur dans l'espace de paramètre :  $\forall \rho, \theta, A(\rho, \theta) = 0$
- ④ Pour tous les points appartenant au contour, on ajoute un vote pour toutes les droites possibles passant par ce point :  $\forall (x, y) \in \text{contour}, \forall \theta, A(x \sin \theta + y \cos \theta, \theta)++$

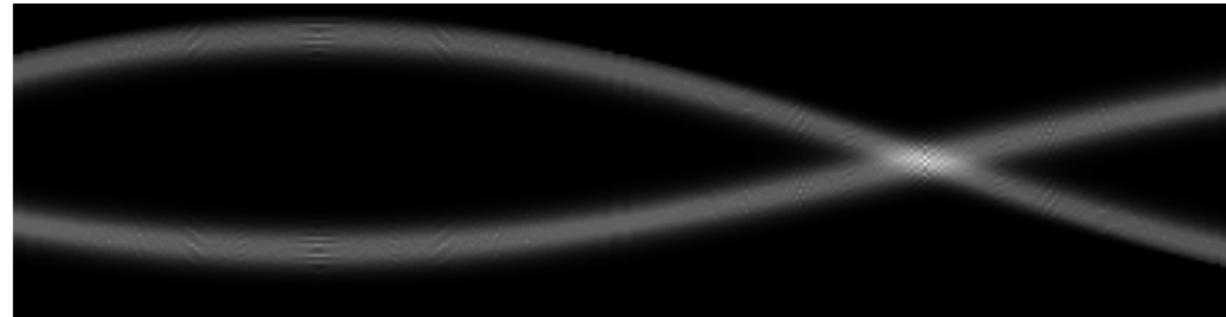


# Transformée de Hough

Algorithme basé sur la détection des contours

## Algorithme

- ① Calcul du gradient
- ② Extraction des contours
- ③ Mise à zéro d'un accumulateur dans l'espace de paramètre :  $\forall \rho, \theta, A(\rho, \theta) = 0$
- ④ Pour tous les points appartenant au contour, on ajoute un vote pour toutes les droites possibles passant par ce point :  $\forall (x, y) \in \text{contour}, \forall \theta, A(x \sin \theta + y \cos \theta, \theta)++$

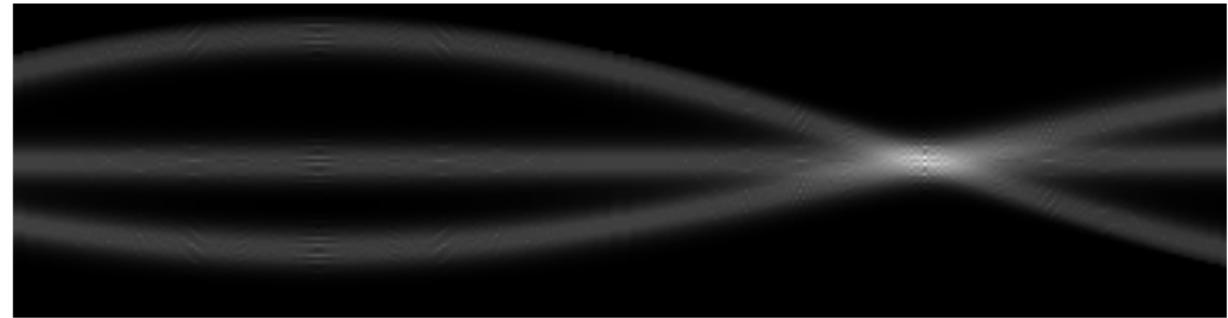
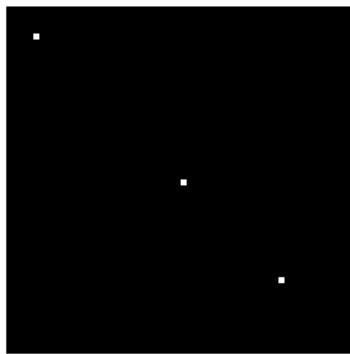


# Transformée de Hough

Algorithme basé sur la détection des contours

## Algorithme

- ① Calcul du gradient
- ② Extraction des contours
- ③ Mise à zéro d'un accumulateur dans l'espace de paramètre :  $\forall \rho, \theta, A(\rho, \theta) = 0$
- ④ Pour tous les points appartenant au contour, on ajoute un vote pour toutes les droites possibles passant par ce point :  $\forall (x, y) \in \text{contour}, \forall \theta, A(x \sin \theta + y \cos \theta, \theta)++$

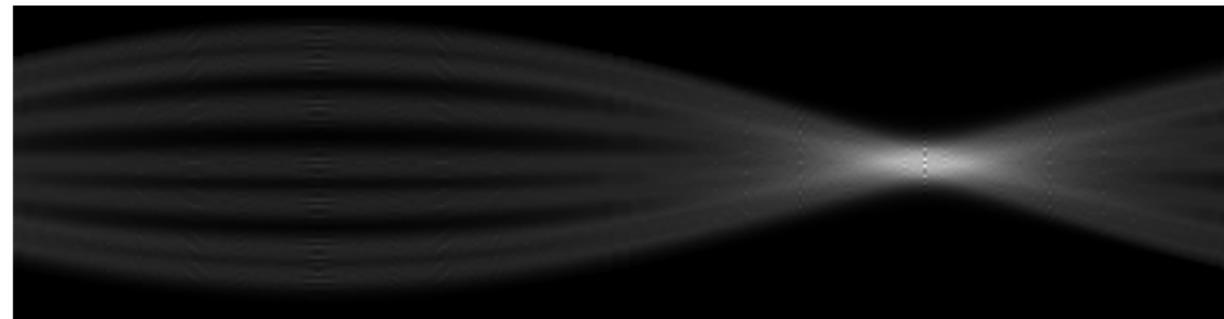
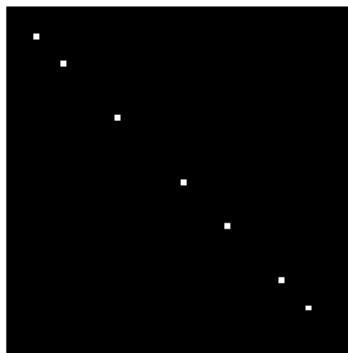


# Transformée de Hough

Algorithme basé sur la détection des contours

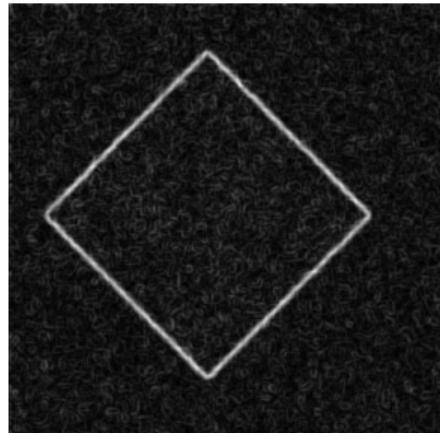
## Algorithme

- ① Calcul du gradient
- ② Extraction des contours
- ③ Mise à zéro d'un accumulateur dans l'espace de paramètre :  $\forall \rho, \theta, A(\rho, \theta) = 0$
- ④ Pour tous les points appartenant au contour, on ajoute un vote pour toutes les droites possibles passant par ce point :  $\forall (x, y) \in \text{contour}, \forall \theta, A(x \sin \theta + y \cos \theta, \theta)++$

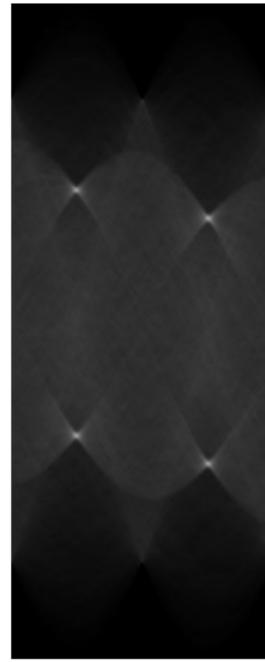


# Transformée de Hough

Exemple simple



Gradient



Transformée de Hough  
/ contours



Transformée de Hough  
/ gradient

# Transformée de Hough

Adaptation pour la recherche de formes circulaires

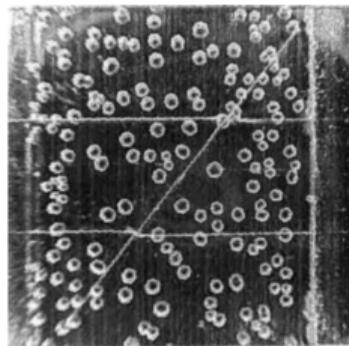
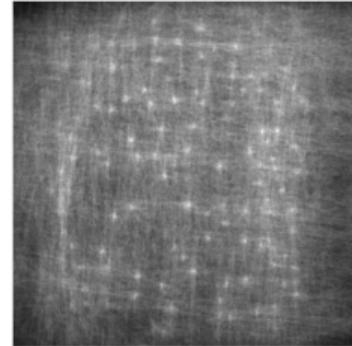
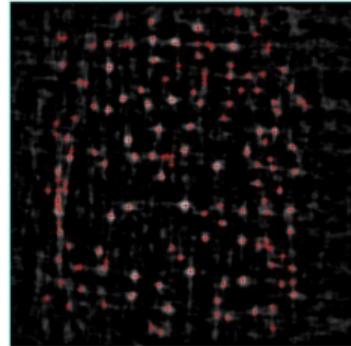


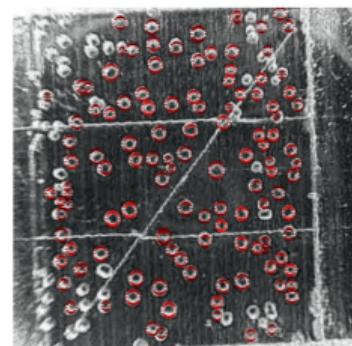
Image d'origine



Accumulation des  
rayons



Filtrage  
passe-bande



Identification des  
cercles

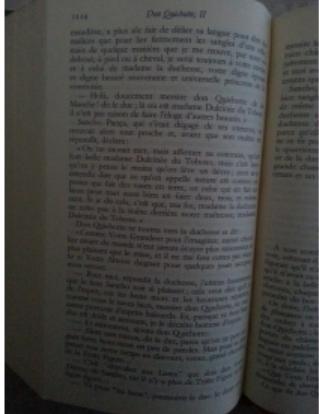
# Plan

- 1 Représentation des images
  - Espaces de couleur classiques
  - Algorithmes de détection simple
- 2 Filtrage
  - Filtrage non linéaire (médian, bilatéral)
  - Filtrage directif
  - Traitements dans le domaine fréquentiel
- 3 Reconnaissance de formes simples
  - Canny
  - Transformée de Hough
- 4 Segmentation d'image
  - Seuillage
- Ligne de partage des eaux
- Superpixels
- Applications : recherche de zones d'intérêt
- 5 Reconnaissance d'objets
  - Détection d'objets génériques
  - Détection de points d'intérêts
- 6 Traitements vidéo
  - Calcul de flux optique
- 7 Vision 3d
- 8 Autres fonctions
- 9 Formations

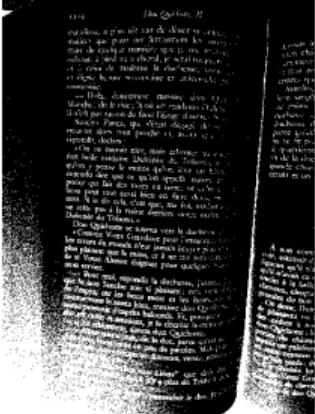
# Segmentation d'image

## Seuillage

- **Principe :** image en niveaux de gris → image binaire
- **Problème :** **choix du seuil.**



Image



Seuillage fixe



Seuillage adaptatif

## Solutions :

- Seuil automatique (fixe) : **Otsu**
- **Seuillage adaptatif**

# Segmentation d'image

## Ligne de partage des eaux

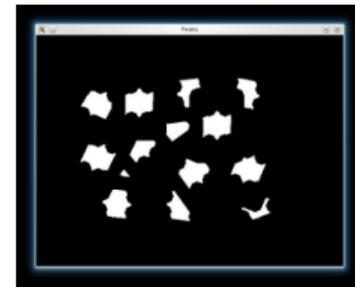
- « *Watershed* »
- Nécessite des points de départ (« graines »), puis remplissage
- Pour trouver les points de départ : **transformée de distance**



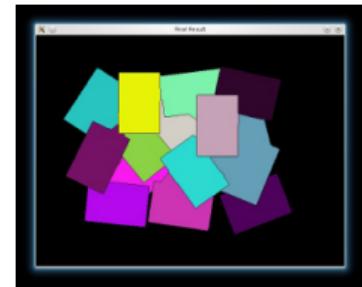
Image



Transformée de  
distance



Seuillage

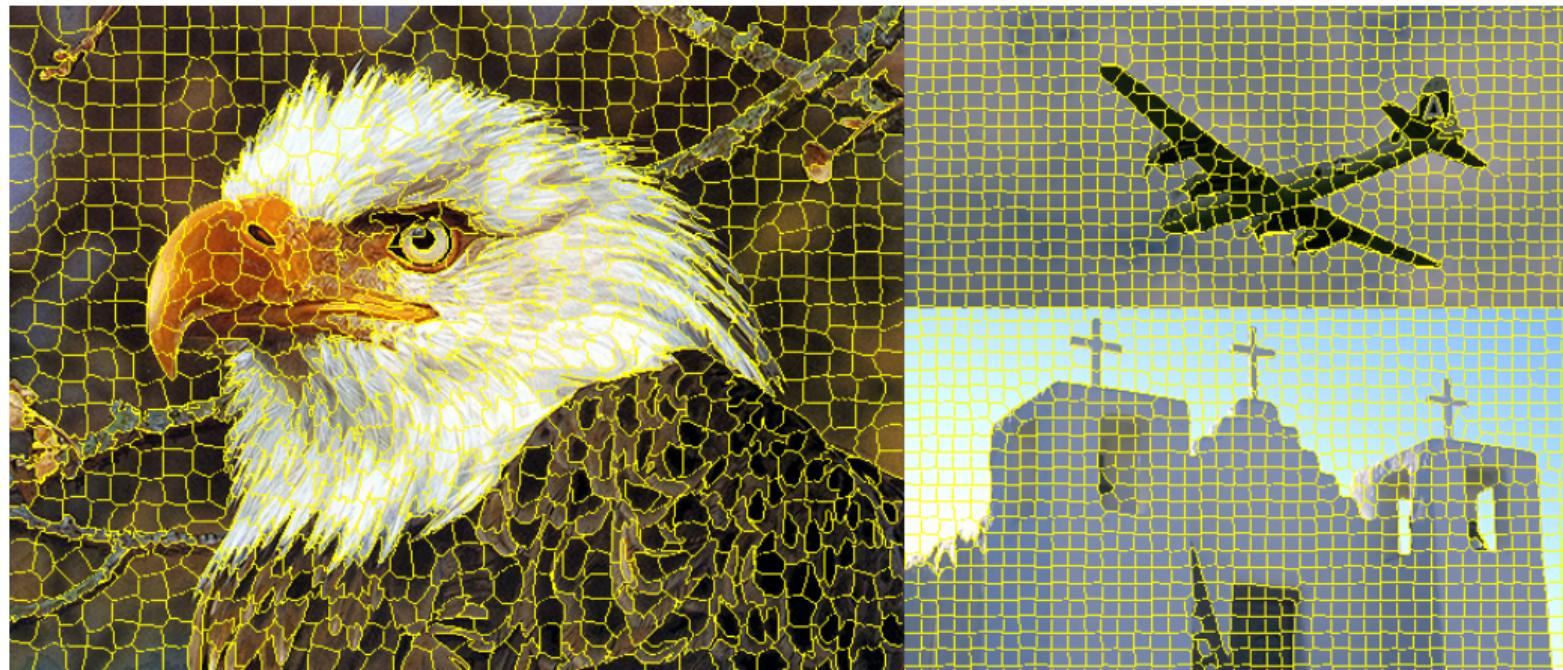


Segmentation

# Segmentation d'image

## Superpixels

- Graines uniformément réparties
- 1 **super-pixel** = zone  $\sim$  uniforme



# Segmentation hiérarchique

Exemple



Niveau 1 (racine)



Niveau 30 (~2500 segments)



Niveau 51 (~8700 segments)



Niveau 91 (~18000 segments)

# Segmentation d'image

Applications : recherche de zones d'intérêt



1 - Image initiale ( $\sim 900$  KPixels)



2 - Segmentation ( $\sim 14$  KSegments)



3 - Zones d'intérêts ( $\sim 10$  zones)



4 - Identification

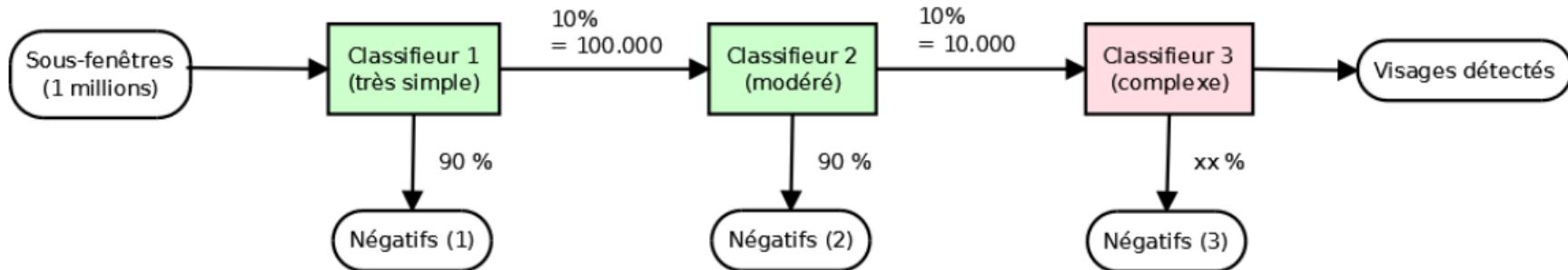
# Plan

- 1 Représentation des images
  - Espaces de couleur classiques
  - Algorithmes de détection simple
- 2 Filtrage
  - Filtrage non linéaire (médian, bilatéral)
  - Filtrage directif
  - Traitements dans le domaine fréquentiel
- 3 Reconnaissance de formes simples
  - Canny
  - Transformée de Hough
- 4 Segmentation d'image
  - Seuillage
- Ligne de partage des eaux
- Superpixels
- Applications : recherche de zones d'intérêt
- 5 Reconnaissance d'objets
  - Détection d'objets génériques
  - Détection de points d'intérêts
- 6 Traitements vidéo
  - Calcul de flux optique
- 7 Vision 3d
- 8 Autres fonctions
- 9 Formations

# Détection d'objets génériques

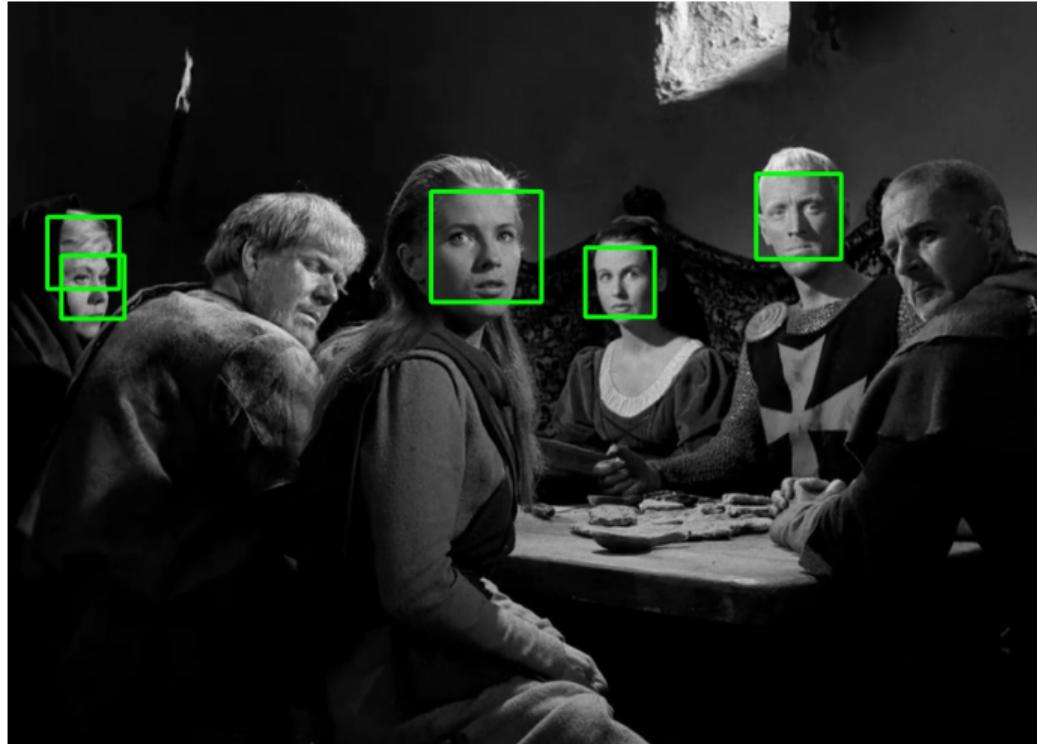
## Principe général

- On suppose que la taille de l'objet à détecter est connue
- Alors on va tester toutes zones d'intérêt possibles dans l'image
- Mais : le nombres de zones possibles est très important  
⇒ Besoin d'éliminer très rapidement les zones qui ne présentent pas d'intérêt
- C'est le principe d'un détecteur en cascade :



# DéTECTEURS EN CASCADE

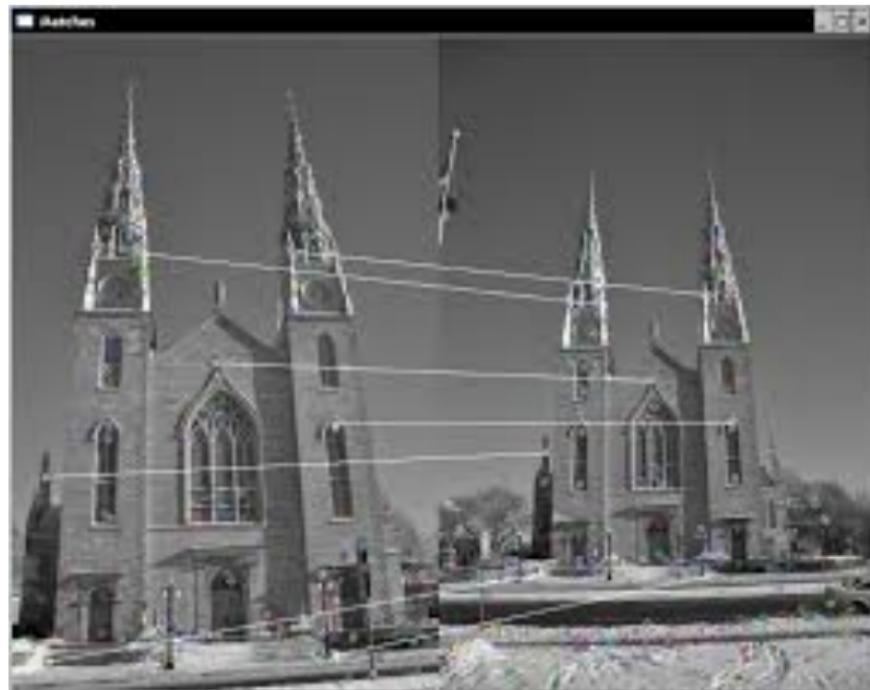
Exemple



# Mise en correspondance de points d'intérêt

## Applications

- Reconnaissance d'une image / objet d'après une base de données (FLANN)
- Suivi d'objets en mouvement
- Réalisation de panoramas
- Estimation d'homographies
- Vision stéréo
- etc.



# Mise en correspondance de points d'intérêts

## ① Localiser des points d'intérêt :

- ▶ En nombre limité
- ▶ Stabilité, répétabilité

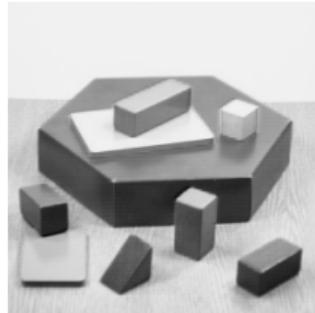
## ② Pour chaque point d'intérêt, **calculer des traits**

## ③ Faire correspondre des points d'intérêt sur 2 images

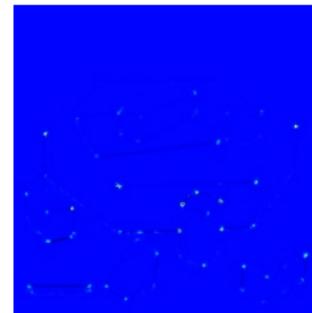


# Détection de points d'intérêts

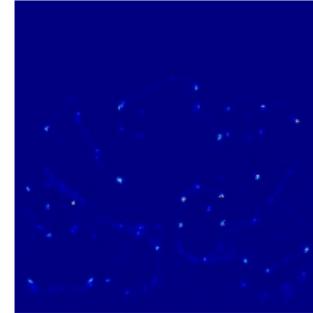
Détecteurs de Harris / Shi-Tomasi



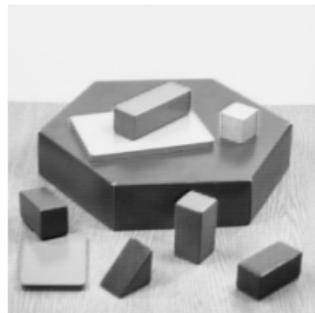
Image



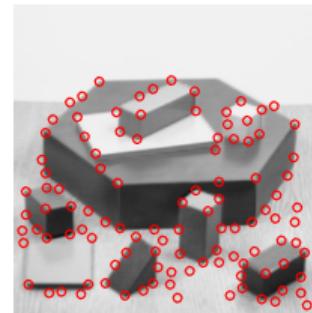
Harris



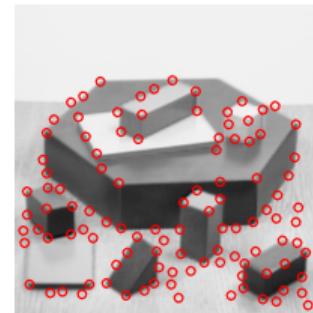
$\min(\lambda)$



Image



GFTT (Harris)

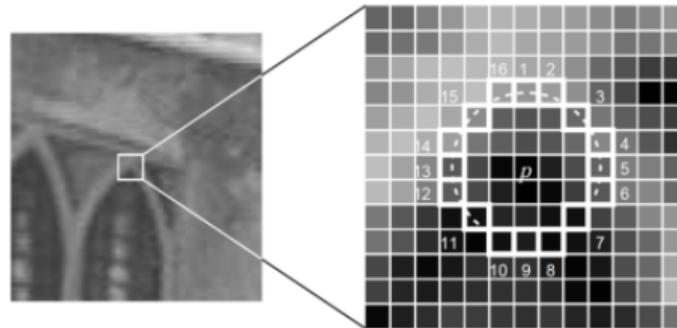


GFTT ( $\min(\lambda)$ )

# Détection de points d'intérêts

## Autres détecteurs

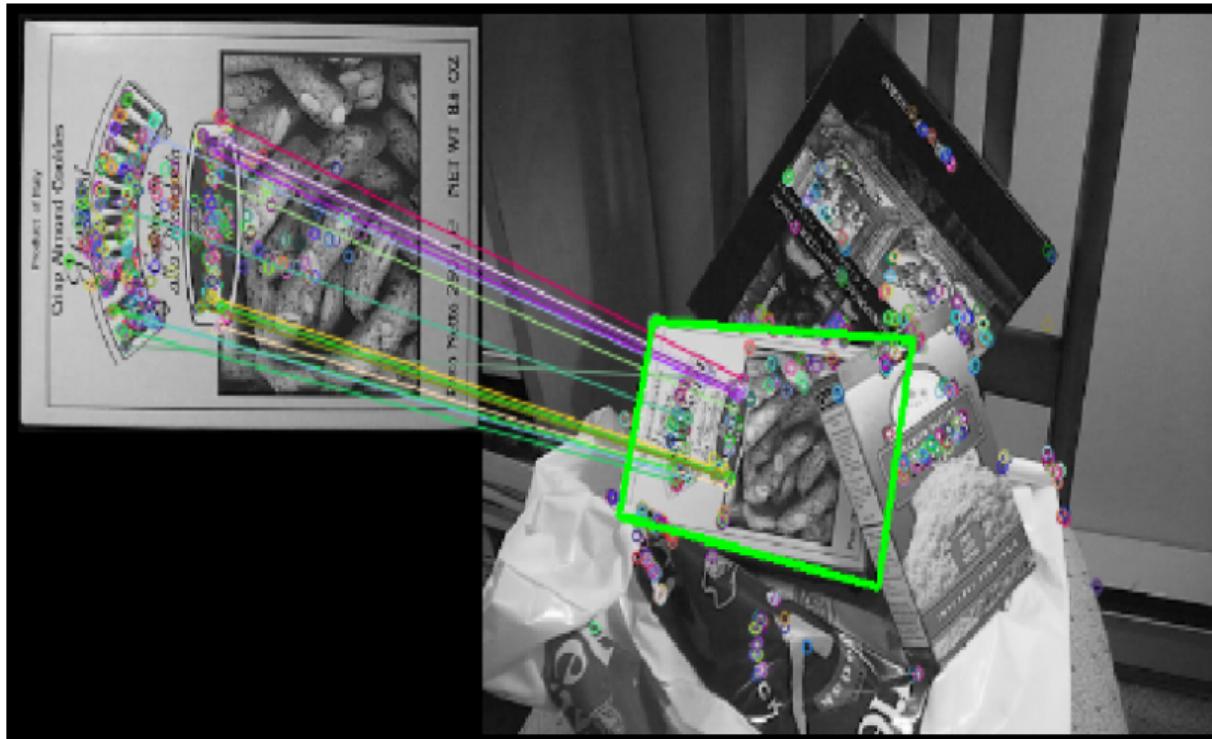
- **FAST** (2006) : Un cercle autour de chaque pixel. Au moins  $n$  pixels dans le cercle bien supérieurs (ou bien inférieurs) au pixel central.



- **ORB** (2011) : Calcul sur une pyramide. Tri des points FAST (FAST-9) d'après le score de Harris et sélection des  $N$  meilleurs coins. Mesure de l'orientation.
- **BRISK** (2011) : Fondé sur FAST avec une pyramide (4 images / octave) et interpolation quadratique. Calcul d'orientation.

# Mise en correspondance

Exemple (avec descripteurs ORB)



# Plan

- 1 Représentation des images
  - Espaces de couleur classiques
  - Algorithmes de détection simple
- 2 Filtrage
  - Filtrage non linéaire (médian, bilatéral)
  - Filtrage directif
  - Traitements dans le domaine fréquentiel
- 3 Reconnaissance de formes simples
  - Canny
  - Transformée de Hough
- 4 Segmentation d'image
  - Seuillage
- Ligne de partage des eaux
- Superpixels
- Applications : recherche de zones d'intérêt
- 5 Reconnaissance d'objets
  - Détection d'objets génériques
  - Détection de points d'intérêts
- 6 Traitements vidéo
  - **Calcul de flux optique**
- 7 Vision 3d
- 8 Autres fonctions
- 9 Formations

# Détection d'arrière-plan

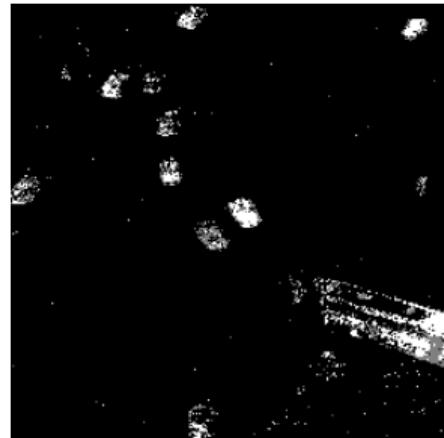
Exemple (mélange de gaussiennes)



Vidéo



Arrière-plan



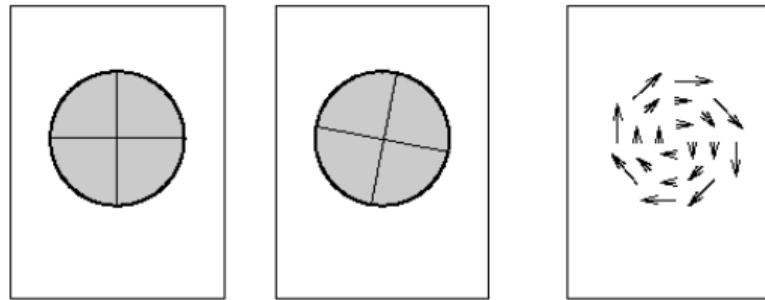
Avant-plan

# Calcul de flux optique

## Principe

- Flux optique = vitesse de chaque pixel sur une séquence vidéo :

$$\phi(t) = \begin{pmatrix} \phi_x(t) & \phi_y(t) \end{pmatrix} = \begin{pmatrix} \frac{dx}{dt} & \frac{dy}{dt} \end{pmatrix}$$



- **Méthodes creuses** (Lucas-Kanade) : le flux optique est calculé seulement au niveau de points d'intérêts
- **Méthodes denses** (Farnebäck, TV L1, SimpleFlow) : sur toute l'image

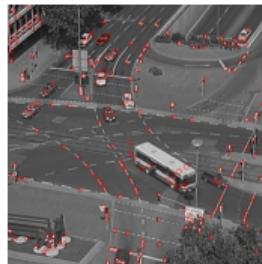
# Calcul de flux optique

## Exemples

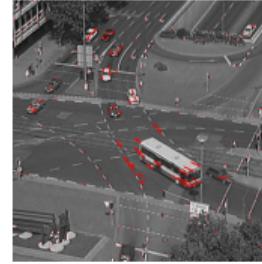
- Flux optique creux :



Vidéo



Points d'intérêt

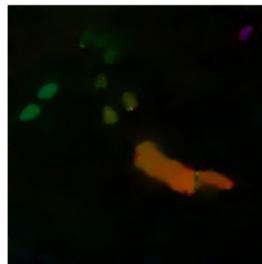


Flux optique

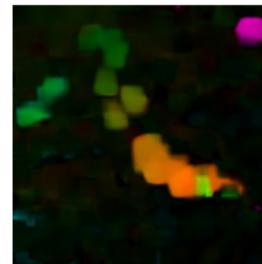
- Flux optique dense :



Vidéo



TV-L1



Farnebäck

# Plan

## 1 Représentation des images

- Espaces de couleur classiques
- Algorithmes de détection simple

## 2 Filtrage

- Filtrage non linéaire (médian, bilatéral)
- Filtrage directif
- Traitements dans le domaine fréquentiel

## 3 Reconnaissance de formes simples

- Canny
- Transformée de Hough

## 4 Segmentation d'image

- Seuillage

## • Ligne de partage des eaux

## • Superpixels

## • Applications : recherche de zones d'intérêt

## 5 Reconnaissance d'objets

- Détection d'objets génériques
- Détection de points d'intérêts

## 6 Traitements vidéo

- Calcul de flux optique

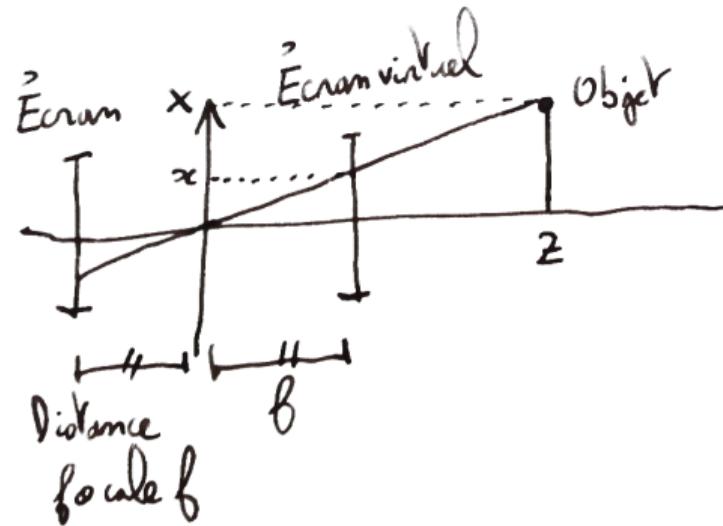
## 7 Vision 3d

## 8 Autres fonctions

## 9 Formations

# Modèles de caméra

Sténopé : projection de perspective (modèle simplifié)



$$\frac{f}{Z} = \frac{x}{X} \Rightarrow x = \frac{Xf}{Z}$$

Plus généralement :  $x = \frac{f_x X}{Z} + u_0, \quad y = \frac{f_y Y}{Z} + v_0$

Et modèle de scène :  $\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = R \begin{pmatrix} X_s \\ Y_s \\ Z_s \end{pmatrix} + t$

# Modèle de caméra

## Modélisation des distorsions

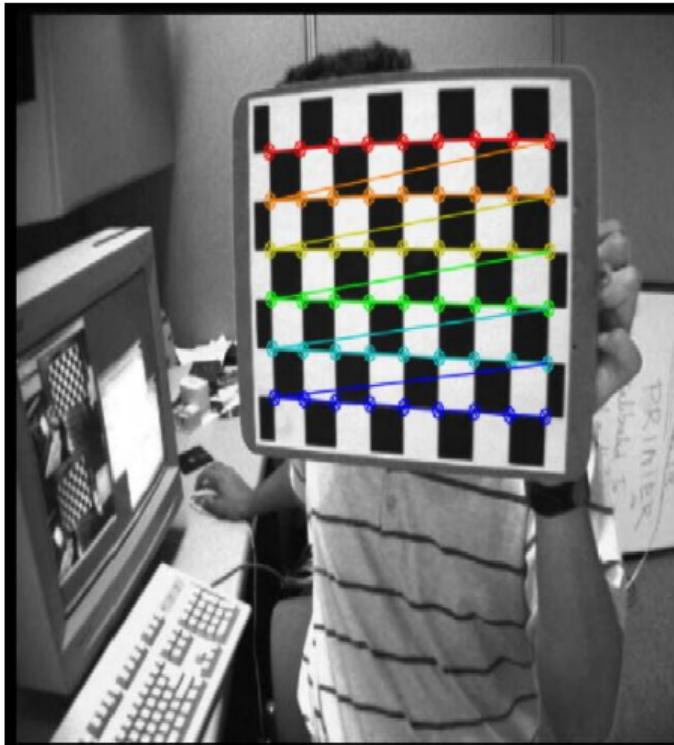
- Les lentilles ne sont pas parfaitement paraboliques (effet « *fisheye* »)



- **Indépendant de la scène** ⇒ correction possible après calibration

# Calibration automatique

Exemple : correction de distorsion



Detection des coins



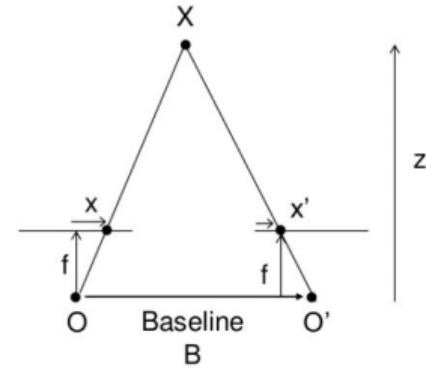
Distortion corrigée

# Vision stéréo

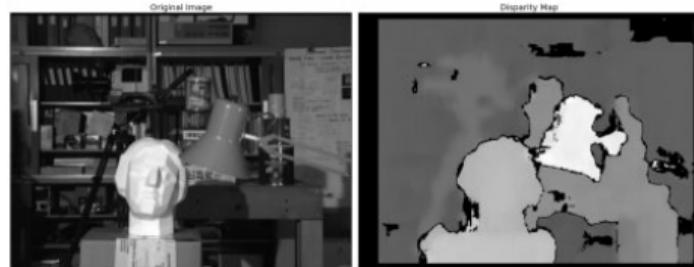
## Estimation de la distance des objets

- Supposons que les caméras soient *parfaitement alignées verticalement et regardant dans la même direction*
- Alors la projection d'un point 3D sur les 2 caméras diffère uniquement en  $x$ , et :

$$x - x' = \frac{Bf}{z}$$



(source : manuel OpenCV)



# Plan

- 1 Représentation des images
  - Espaces de couleur classiques
  - Algorithmes de détection simple
- 2 Filtrage
  - Filtrage non linéaire (médian, bilatéral)
  - Filtrage directif
  - Traitements dans le domaine fréquentiel
- 3 Reconnaissance de formes simples
  - Canny
  - Transformée de Hough
- 4 Segmentation d'image
  - Seuillage
- Ligne de partage des eaux
- Superpixels
- Applications : recherche de zones d'intérêt
- 5 Reconnaissance d'objets
  - Détection d'objets génériques
  - Détection de points d'intérêts
- 6 Traitements vidéo
  - Calcul de flux optique
- 7 Vision 3d
- 8 Autres fonctions
- 9 Formations

## Quelques autres fonctions

- Stabilisation vidéo
- Super-résolution



- Fonctions **d'apprentissage automatique** (KNN, SVM, Adaboost, etc.)
- *Image stitching* (panoramas)
- Module photo : débruitage, fusion HDR, etc.
- Accélérations GPU : CUDA (NVIDIA), OpenCL (portable)
- Nombreuses contributions

# Plan

- 1 Représentation des images
  - Espaces de couleur classiques
  - Algorithmes de détection simple
- 2 Filtrage
  - Filtrage non linéaire (médian, bilatéral)
  - Filtrage directif
  - Traitements dans le domaine fréquentiel
- 3 Reconnaissance de formes simples
  - Canny
  - Transformée de Hough
- 4 Segmentation d'image
  - Seuillage
- 5 Ligne de partage des eaux
  - Superpixels
  - Applications : recherche de zones d'intérêt
- 6 Reconnaissance d'objets
  - Détection d'objets génériques
  - Détection de points d'intérêts
- 7 Traitements vidéo
  - Calcul de flux optique
- 8 Vision 3d
- 9 Autres fonctions
- 9 Formations

# Formations

<b>OpenCV pour le traitement d'image</b>	Traitement d'image classique (filtrage, reconnaissance de contours, de formes géométriques, etc.), reconnaissance d'objets, points d'intérêts, traitement vidéos (soustraction d'arrière-plan, flux optique), traitements 3d	3 jours
<b>Apprentissage automatique avec OpenCV (2019)</b>	Approche classique : calcul de traits + classifieur. Traits de Haar, HoG, LBP. Technique ACP. Exemple d'apprentissage avec KNN, SVM, ANN simples.	2 jours

# Traitement d'image avec OpenCV

**Merci !**