

Computer Vision

Raphaël Viards – Magellium – Imagerie et Applications

raphael.viards@magellium.fr

Partie III: Détection et Localisation

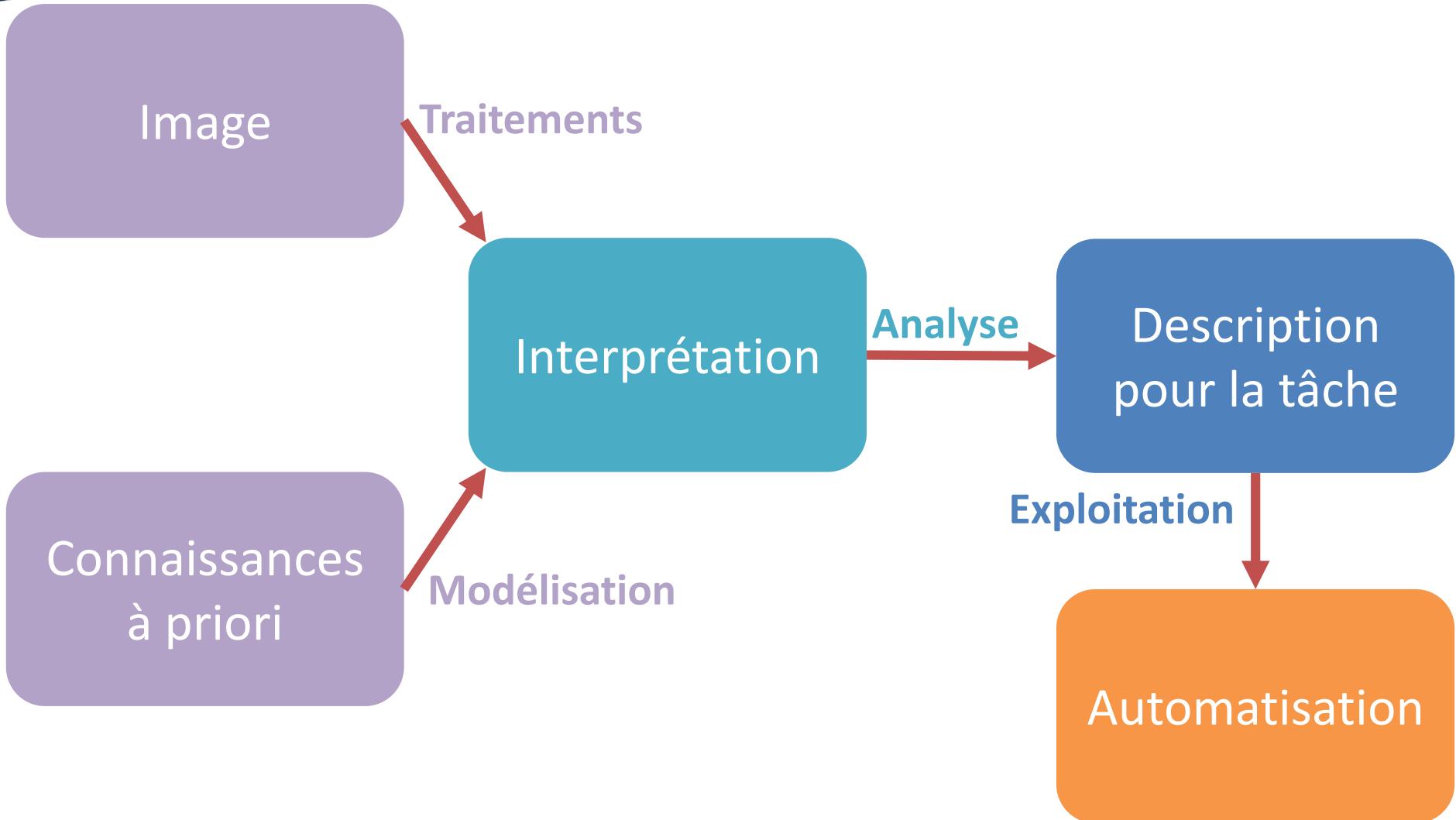
- I. Introduction
- II. Géométrie projective et homographies
- III.Extraction de caractéristiques locales
- IV. Reconnaissance d'objets
- V. Estimation du mouvement

- Qu'est-ce que la vision?
 - Percevoir le monde
 - Composé d'objets
 - Structuré en 3D
 - Efficacement interprété par l'Homme
 - Recueil d'information
 - Ensembles de points
 - Information sur la lumière
 - Quantité et contenu spectral
 - Représentation du monde réel
 - Les objets n'existent pas sur la rétine
 - Processus visuel d'interprétation

- Vision Humaine
 - Extrêmement complexe
 - Active de nombreuses zones du cerveau
 - Possède des capacités nombreuses et variées
- Vision par ordinateur
 - Bio inspirée ou non
 - Production d'un modèle algorithmique fonctionnellement similaire aux capacités du cerveau humain
 - Reproduit seulement un sous-ensemble de capacités

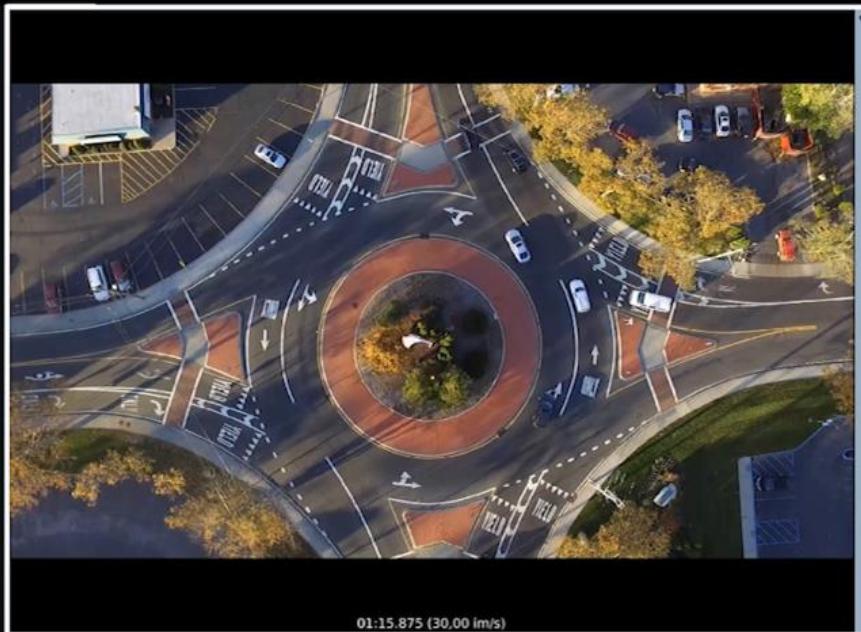
- Quelques termes
 - Traitement d'images
 - Manipulation dont l'entrée et la sortie sont des images
 - Aide l'humain ou la machine à examiner des images
 - Analyse d'images
 - Analyse où l'entrée est une image mais la sortie est une information
 - Vision par ordinateur
 - Émule la vision humaine dans le but d'extraire de l'information ou de prendre une décision
 - Une des branches du domaine de l'intelligence artificielle

Processus d'intégration dans un système



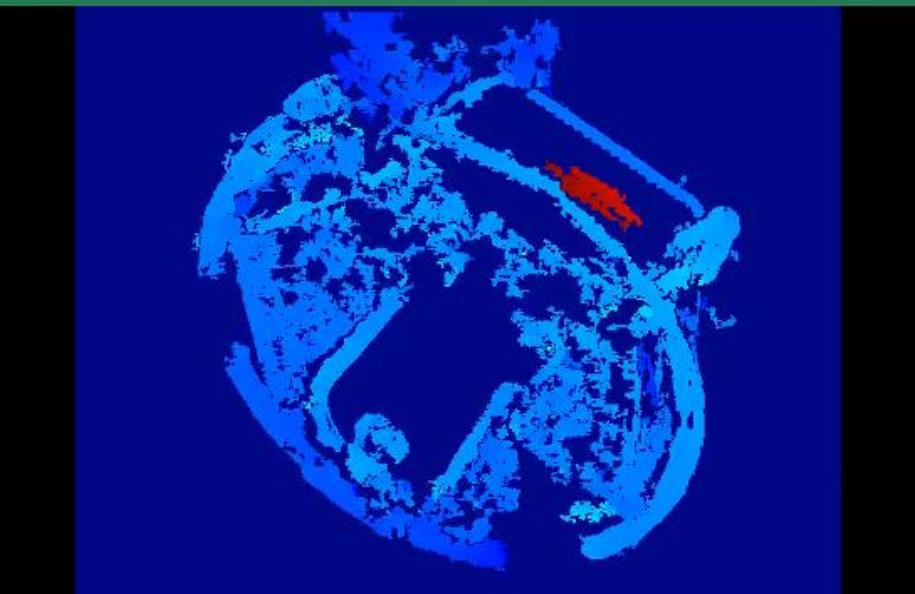
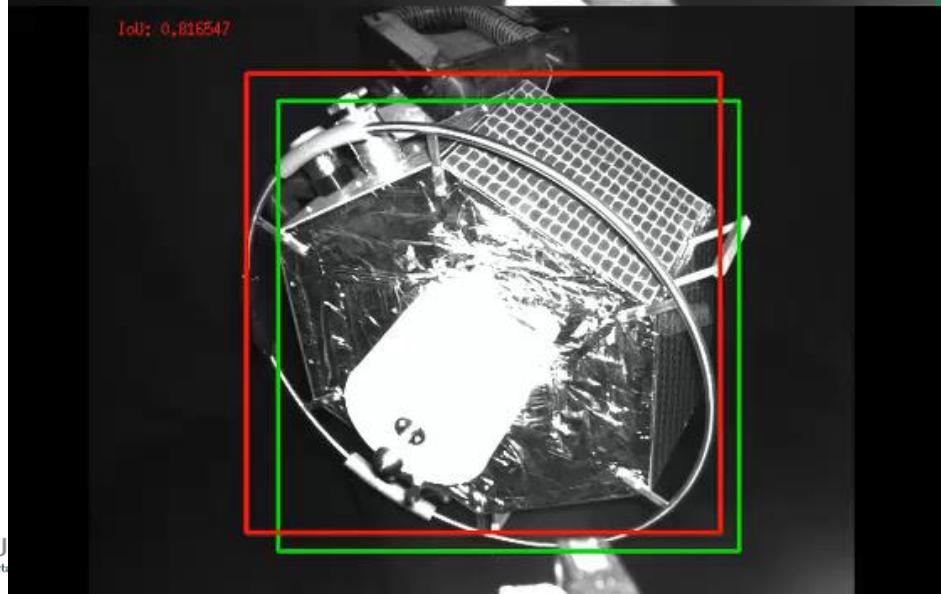
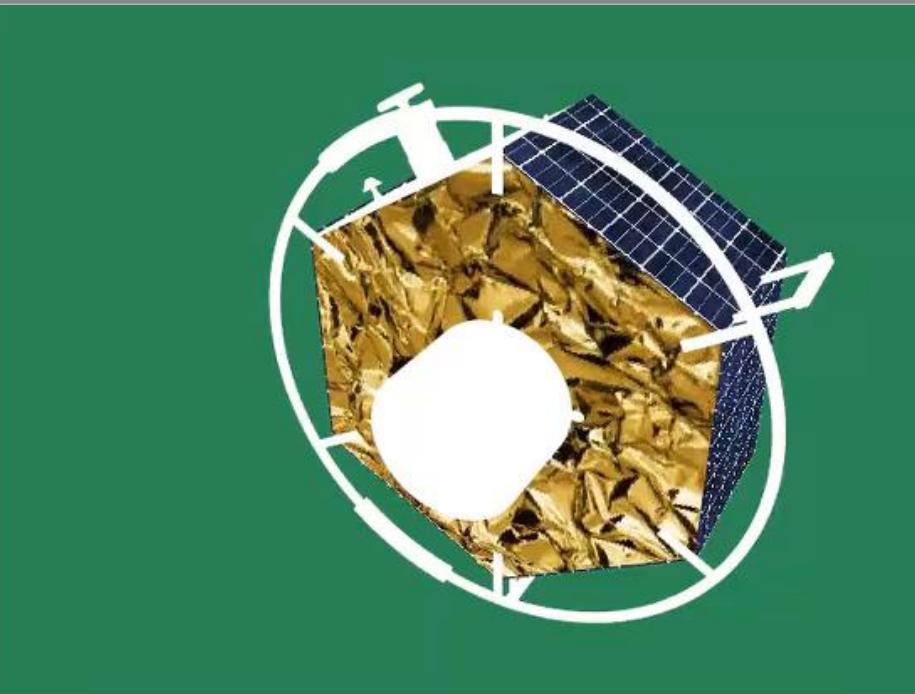
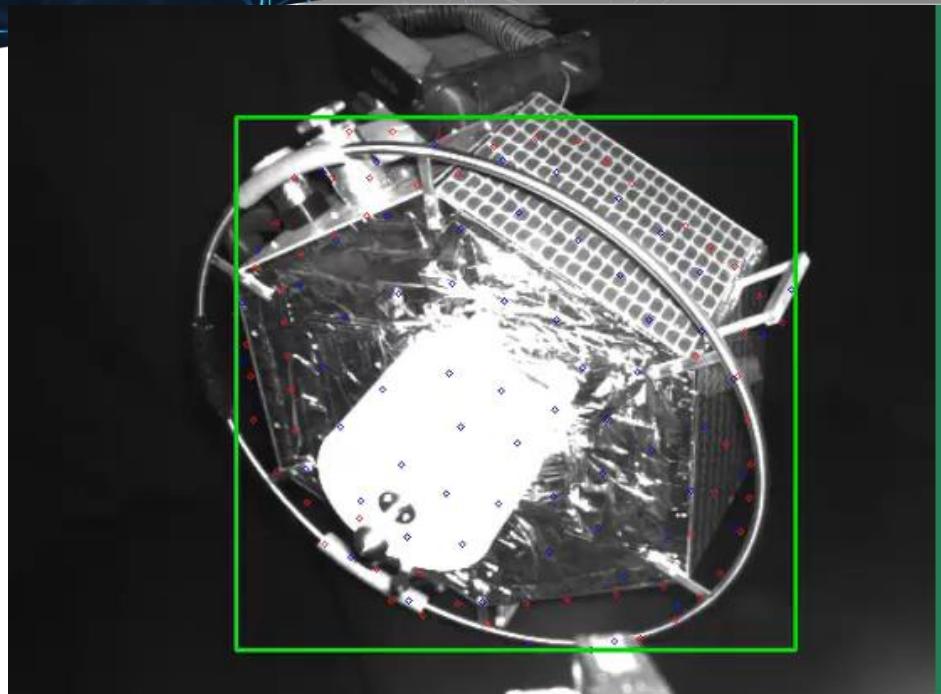
- Objectifs de la séance
 - Trouver/extraire dans l'image des informations pertinentes pour:
 - Reconnaître et détecter des objets
 - Obtenir la position 3D d'objets
 - Cartographier l'environnement
 - Estimer le mouvement

Détection Deep-learning

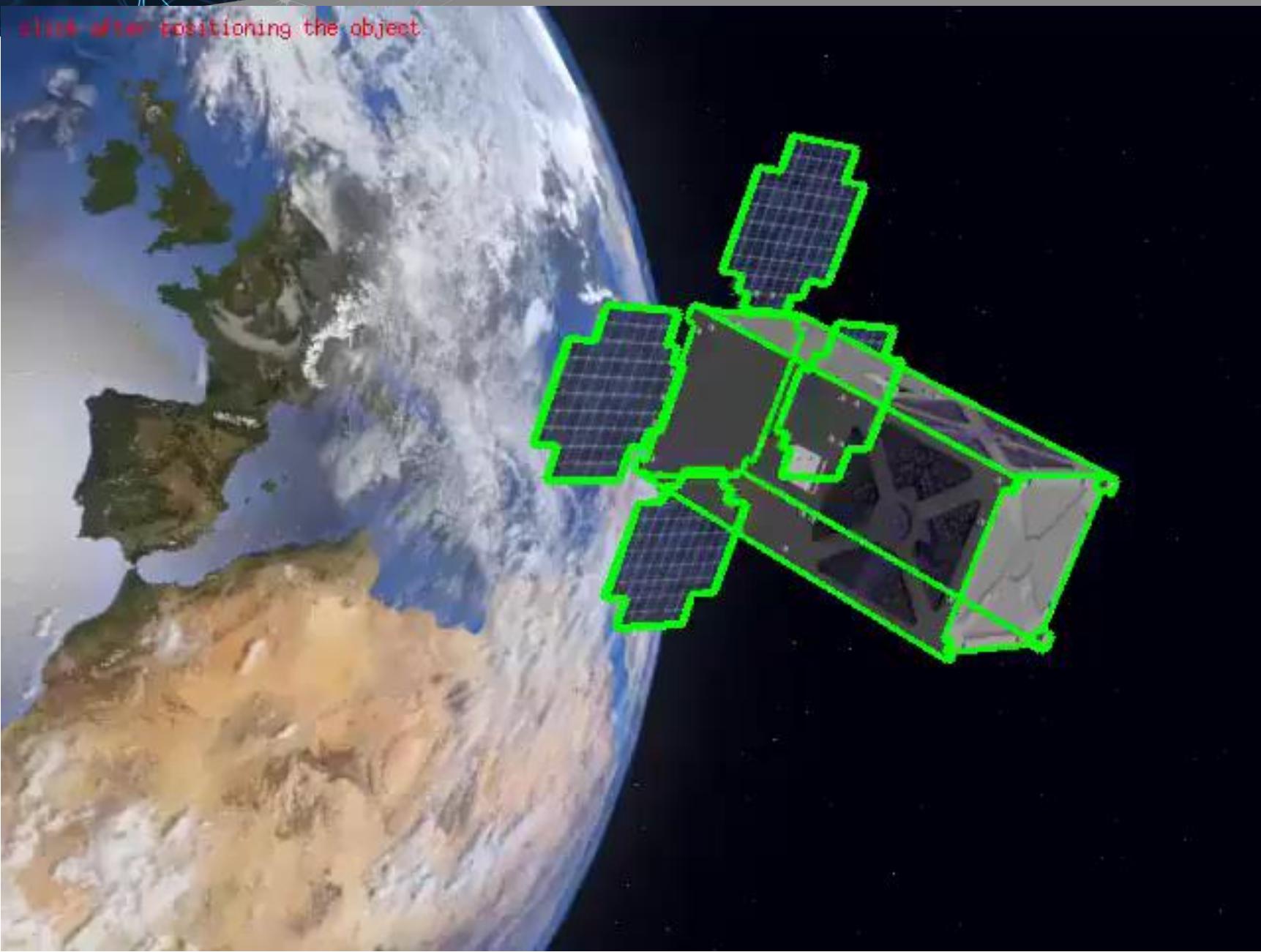


Convolutional Neural Network
Pixel-based labelisation with network confidence

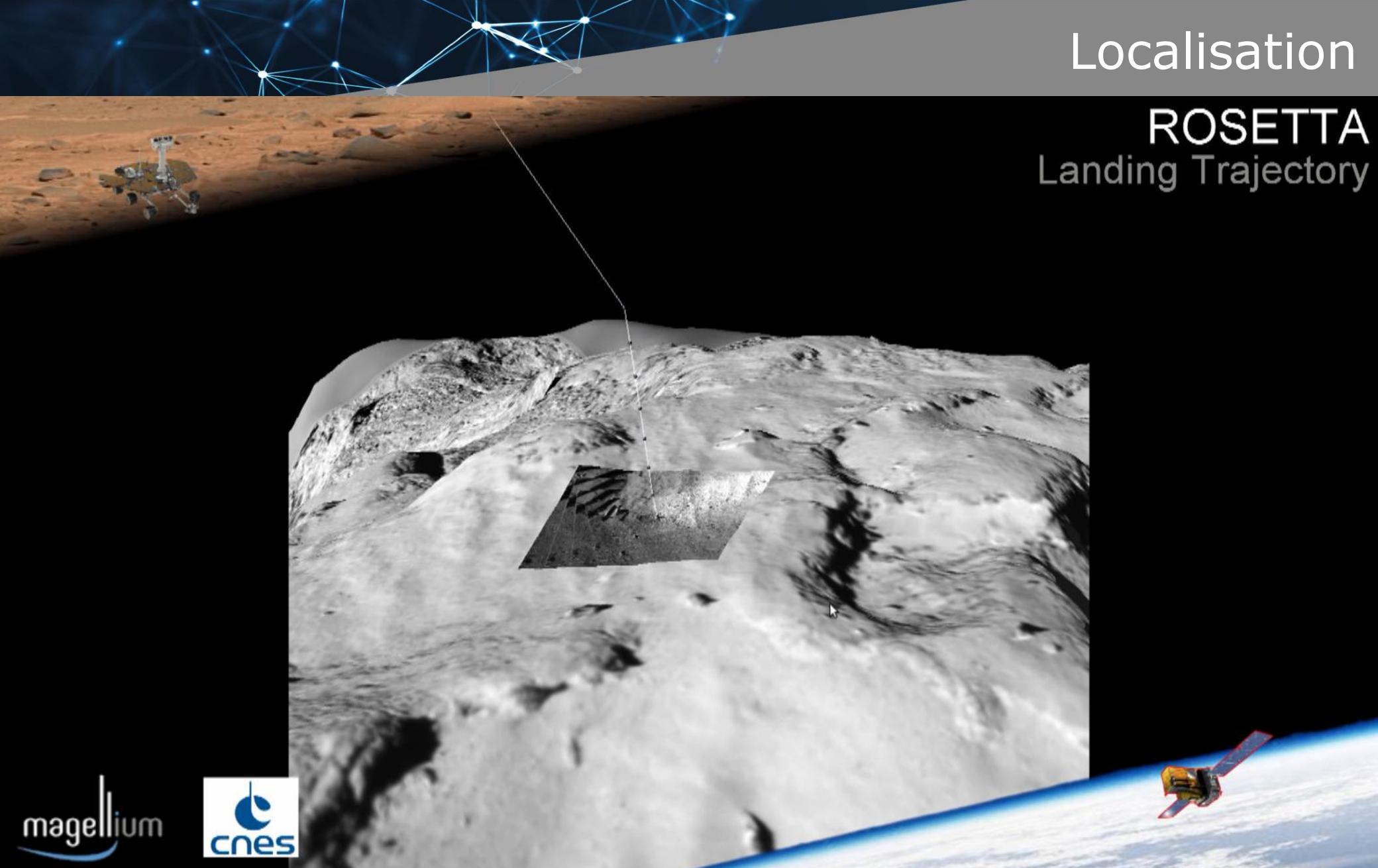
Détection et Tracking



Model-based Tracking



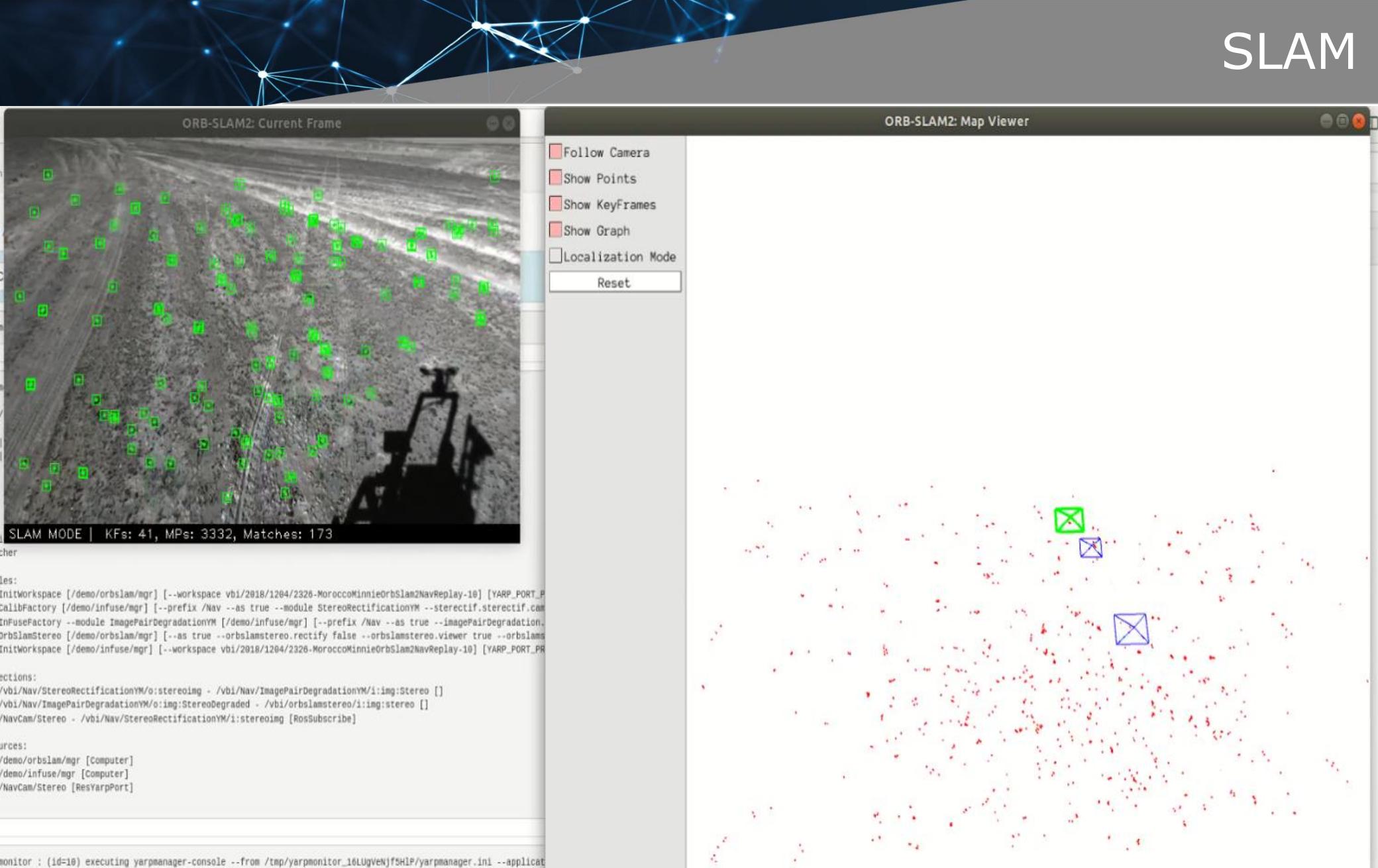
Localisation ROSETTA Landing Trajectory



magellium

cnes

magellium
aerial group



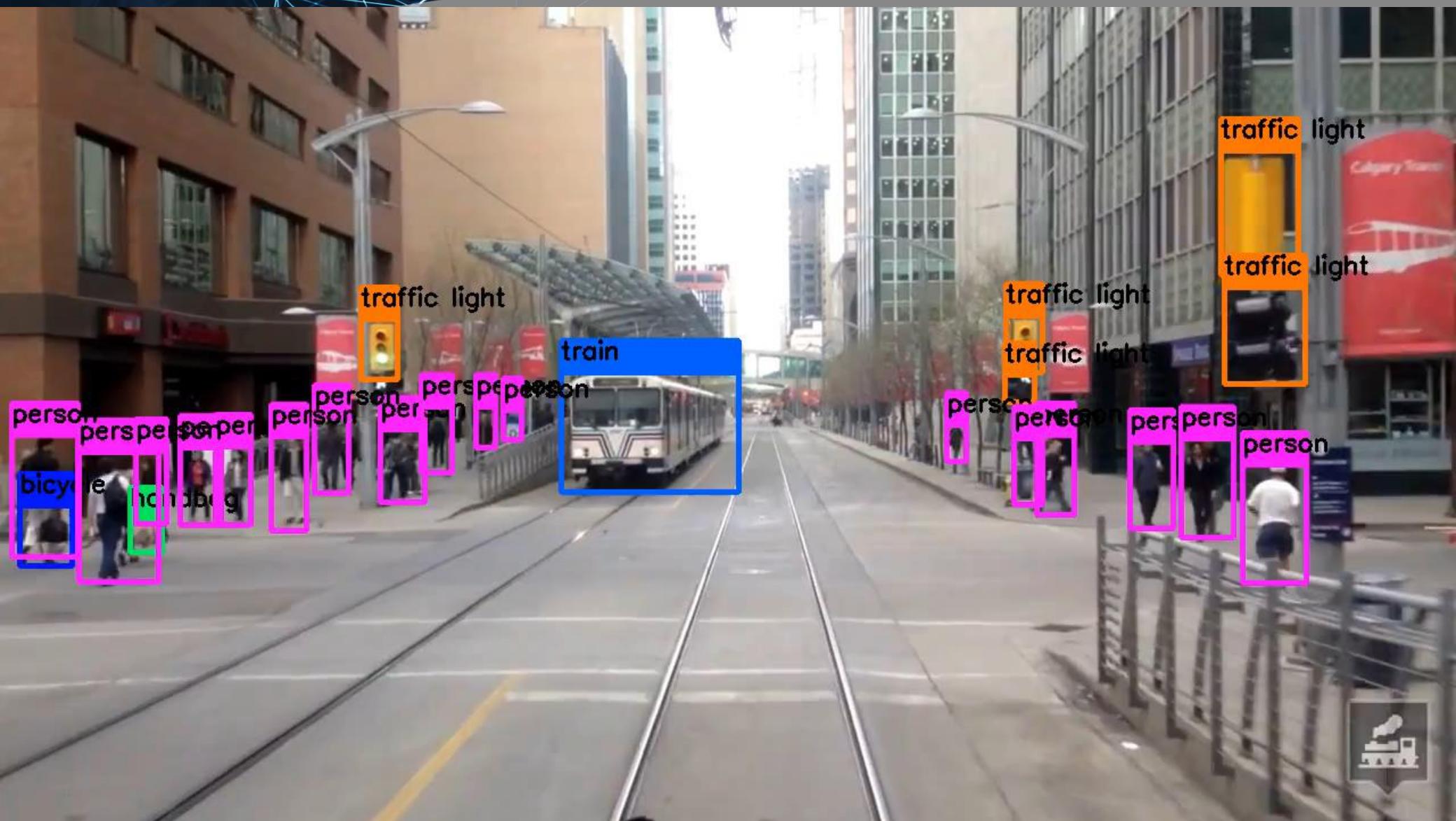


Détection de posture

M31-R



Segmentation sémantique



Partie III: Détection et Localisation

- I. Introduction
- II. Géométrie projective
- III.Extraction de caractéristiques locales
- IV. Reconnaissance d'objets
- V. Estimation du mouvement

- Système de coordonnées pour la géométrie projective
- Passer des coordonnées cartésiennes aux coordonnées homogènes:

$$-\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Passer des coordonnées homogènes aux coordonnées cartésiennes:

$$-\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} u/w \\ v/w \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

- Translation en coordonnées homogènes:

$$- T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbb{I} & t \\ 0 & 1 \end{bmatrix}$$

$$- \bar{x}' = T \cdot \bar{x} \text{ avec } \bar{x} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \text{ et } \bar{x}' = \begin{bmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{bmatrix}$$

- Rotation:

$$- R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow \bar{x}'' = R \cdot \bar{x}$$

- Transformation de corps rigide:

$$- T_r = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

- Autres:

$$- \text{Homothéties: } \Lambda = \begin{bmatrix} \lambda \mathbb{I} & 0 \\ 0 & 1 \end{bmatrix}$$

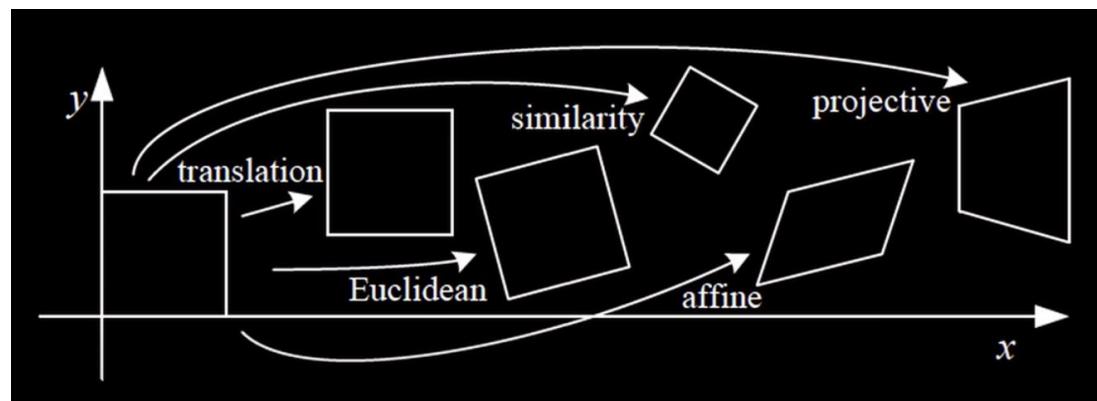
$$- \text{Similitudes: } S = \begin{bmatrix} \lambda R & t \\ 0 & 1 \end{bmatrix}$$

$$- \text{Affinité: } \mathcal{A} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix}$$

$$- \text{Homographie: } P = \begin{bmatrix} A & t \\ p & 1 \end{bmatrix}$$

- Composition de transformées:

$$- \forall n, \exists H \text{ t. q. } H = \prod_{i=0}^{i=n} P_i \text{ avec } \bar{x}' = \left(\prod_{i=0}^{i=n} P_i \right) \cdot \bar{x} = H \cdot \bar{x}$$



- Propriété homogène:

- $\bar{x} \sim \lambda \cdot \bar{x}, \forall \lambda \in \mathbb{R}, \lambda \neq 0$

- Point à l'infini:

- $\bar{x}_{inf} = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}$

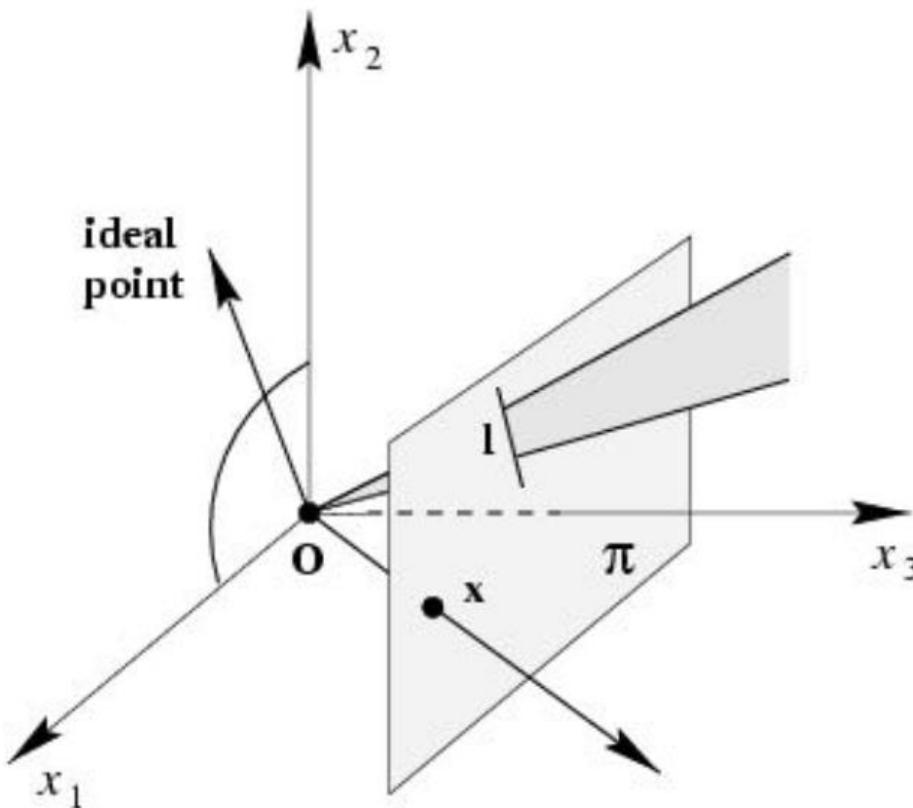
- Représentation homogène des lignes

- $ax + by + c = 0 \Rightarrow l = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$

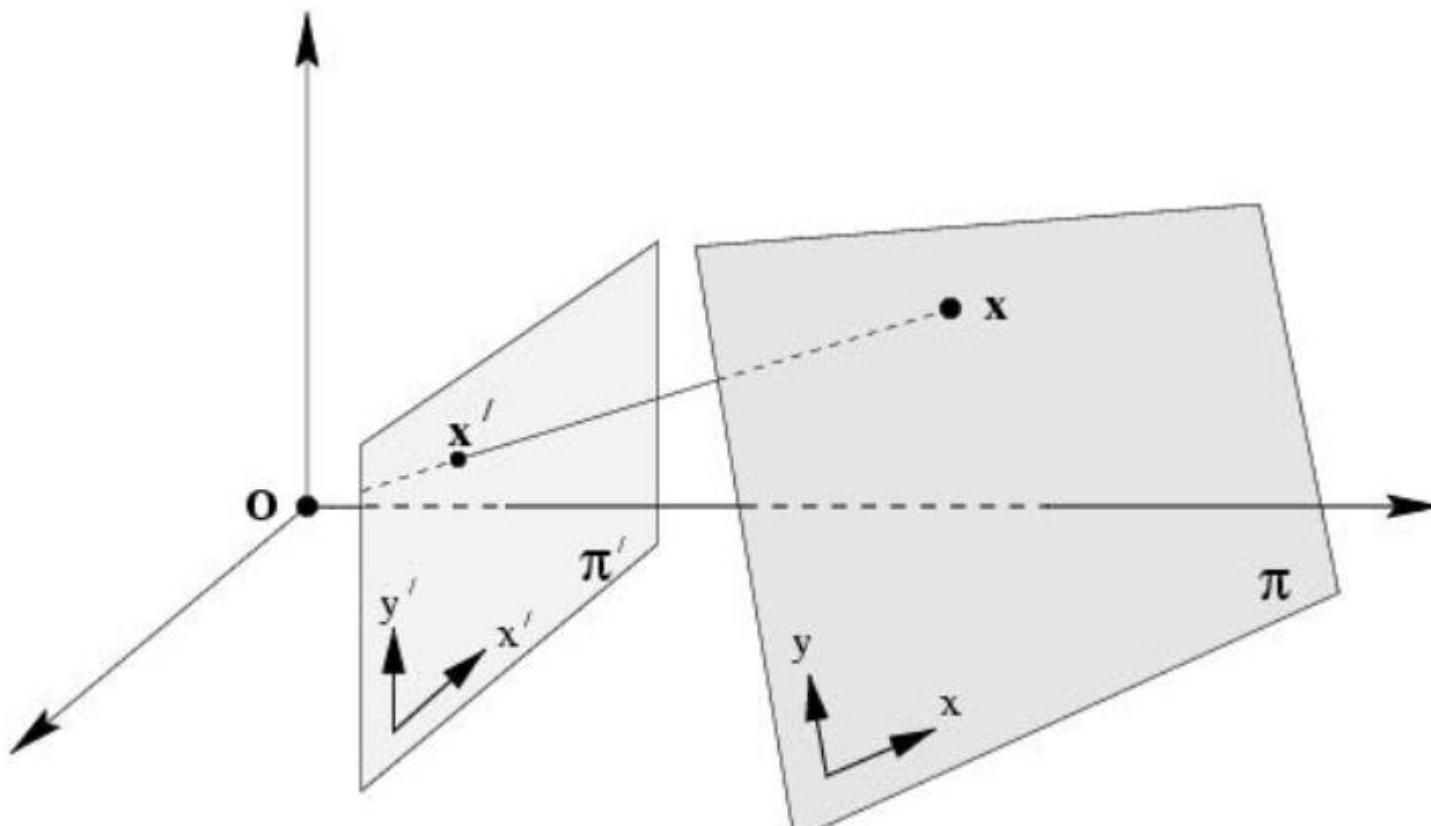
- Appartenance d'un point à une droite:

- $\bar{x} \in l \text{ssi } ax + by + c = 0 \Leftrightarrow \bar{x}^T \cdot l = l^T \cdot \bar{x}$

- Le plan projectif P^2 représente l'espace 3D projeté sur le plan



- Interprétation géométrique de l'homographie



Projection exprimée par $x' = Hx$

- **Définition:**

- Une transformation projective est une bijection h de P^2 dans P^2 telle que 3 points $\bar{x}_1, \bar{x}_2, \bar{x}_3$ sont sur la même droite si et seulement si $h(\bar{x}_1), h(\bar{x}_2), h(\bar{x}_3)$ le sont.
- Théorème:
 - Une bijection $h: P^2 \rightarrow P^2$ est une homographie ssi il existe une matrice 3×3 inversible H telle que pour tout point de P^2 représenté par un vecteur \bar{x} on a $h(\bar{x}) = H \cdot \bar{x}$
- Écriture:

$$\begin{aligned} & \bullet \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} \lambda x' \\ \lambda y' \\ \lambda \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \end{aligned}$$

- Coordonnées homogènes permettent de fixer $h_{33} = 1$
- On a 8 degrés de liberté, 8 paramètres à estimer.

- On a:

- $\bar{x}' = H \cdot \bar{x}$

- $\Rightarrow \lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

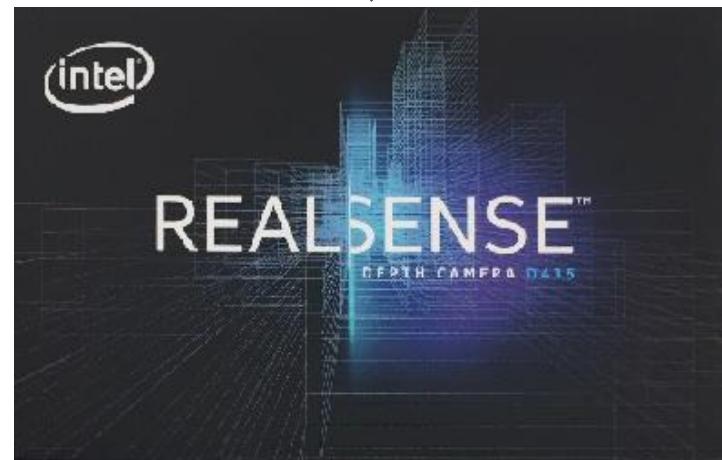
- On pose:

- $Ah = 0$

- Avec $A = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -xx' & -yx' & -x' \\ 0 & 0 & 0 & x & y & 1 & -xy' & -yy' & -y' \end{bmatrix}$

- Et $h = [h_{11} \ h_{12} \ h_{13} \ h_{21} \ h_{22} \ h_{23} \ h_{31} \ h_{32} \ 1]^T$

- Estimation par Direct Linear Transform
 - Nécessite au moins 4 points pour obtenir une solution exacte (2 équations par point et 8 inconnues)
 - Étant donné $n \geq 4$ correspondances de points 2D, déterminer H telle que $\bar{x}'_i = H\bar{x}_i$
- Algorithme:
 - Pour chaque correspondance $\bar{x}_i \leftrightarrow \bar{x}'_i$, calculer A_i
 - Assembler les matrices A_i en une matrice 9×9 : A
 - Calculer le SVD (décomposition en valeurs singulières) de A : $A = U\Sigma V$
 - Solution pour h : dernière colonne de V
 - Déterminer H à partir de h



- Rappel matrice caméra:

$$-\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

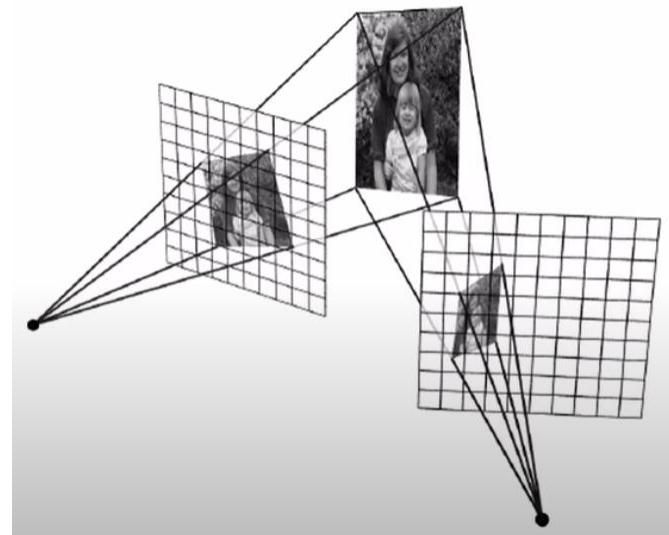
- Supposons que tous les points 3D sont sur un plan:

$$- aX + bY + cZ + d = 0$$

$$-\Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ \frac{aX+bY+d}{-c} \\ 1 \end{bmatrix}$$

Homographie

$$-\Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \boxed{\begin{bmatrix} m'_{11} & m'_{12} & 0 & m'_{14} \\ m'_{21} & m'_{22} & 0 & m'_{24} \\ m'_{31} & m'_{32} & 0 & m'_{34} \end{bmatrix}} \cdot \begin{bmatrix} X \\ Y \\ \frac{aX+bY+d}{-c} \\ 1 \end{bmatrix}$$

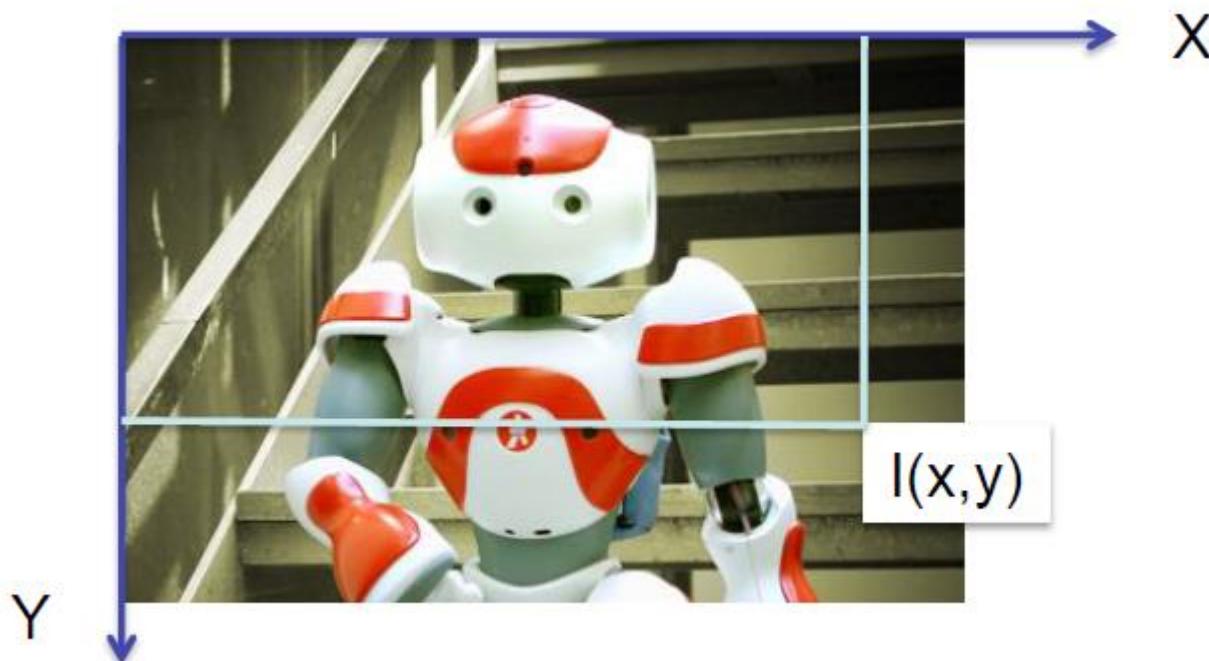


- Le passage de n'importe quel plan vers n'importe quel autre plan (y compris le plan image) est une homographie!

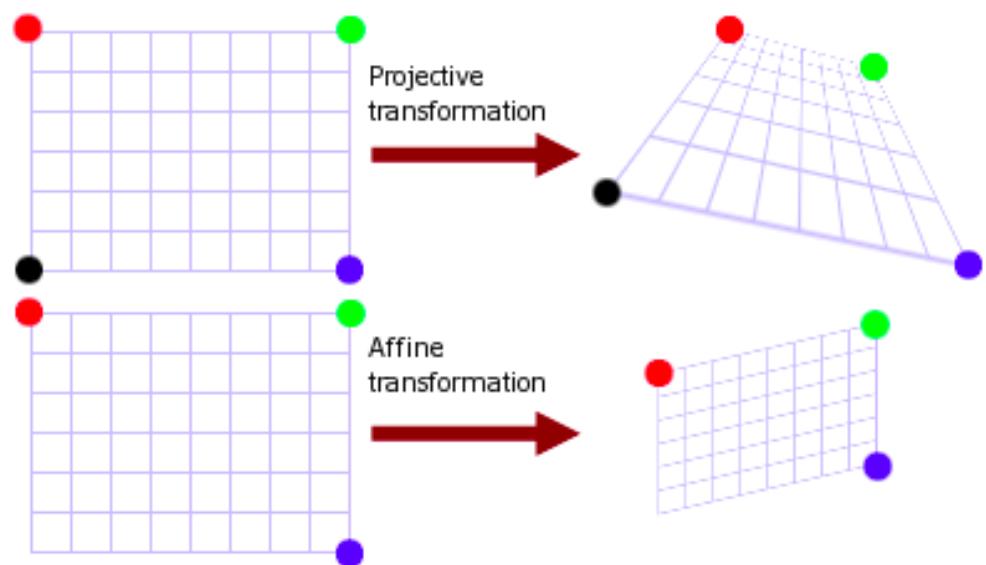
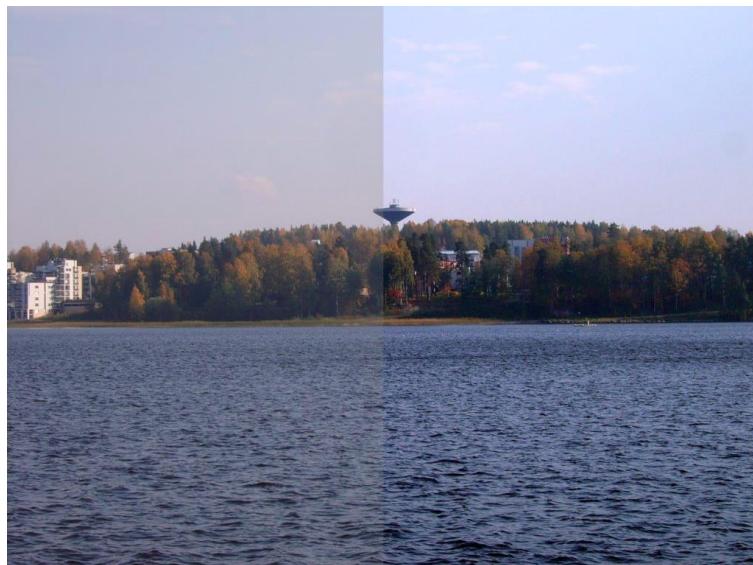
Partie III: Détection et Localisation

- I. Introduction
- II. Géométrie projective
- III.Extraction de caractéristiques locales
- IV. Reconnaissance d'objets
- V. Estimation du mouvement

- $I(x, y)$: valeur d'un pixel
 - Dans \mathbb{R} en monochrome (grayscale)
 - Dans \mathbb{R}^3 en couleur



- De luminosité globale: $I(x, y) \rightarrow I(x, y) + \alpha$
- De contraste: $I(x, y) \rightarrow \lambda I(x, y)$
- Par translation: $I(x, y) \rightarrow I(x + a, y + b)$
- Par transformation affine: $I(x, y) \rightarrow A \cdot I(x, y) + b$
- Par transformation projective: $I(x, y) \rightarrow H \cdot I(x, y)$



- Trouver le point le plus similaire



- Stéréo-Vision: On suppose que les points similaires sont sur la même ligne



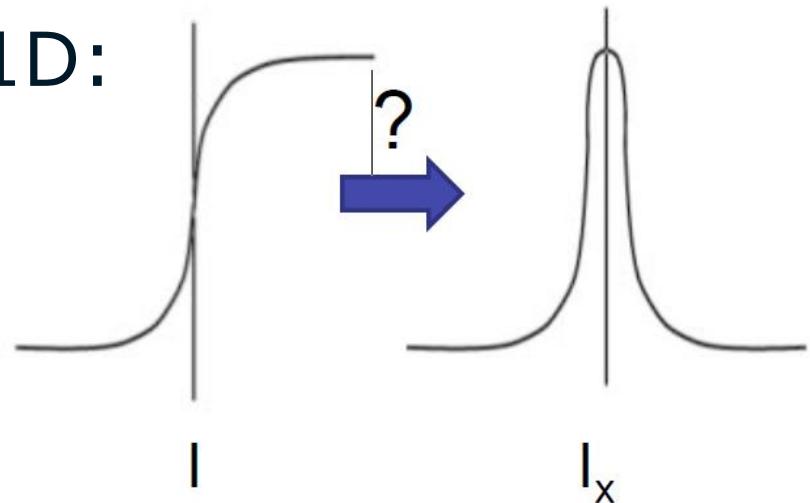
- Comment trouver des points facilement identifiables?
 - Gradients
 - Contours
 - Coins
 - Descripteurs
 - ...

- Aussi appelé noyau ou masque ou encore matrice de convolution
- Permet d'appliquer une opération à l'image
- Convolution: **Kernel**

$$- I_{conv} = \left(\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \right)$$

$$\begin{aligned} - I_{conv}(2, 2) &= 1a + 2b + 3c + 4d + 5e + 6f + 7g + 8h + 9i \\ - I_{conv}(3, 1) &= 0a + 4b + 5c + 0d + 7e + 8f + 0g + 0h + 0i \end{aligned}$$

- En 1D:



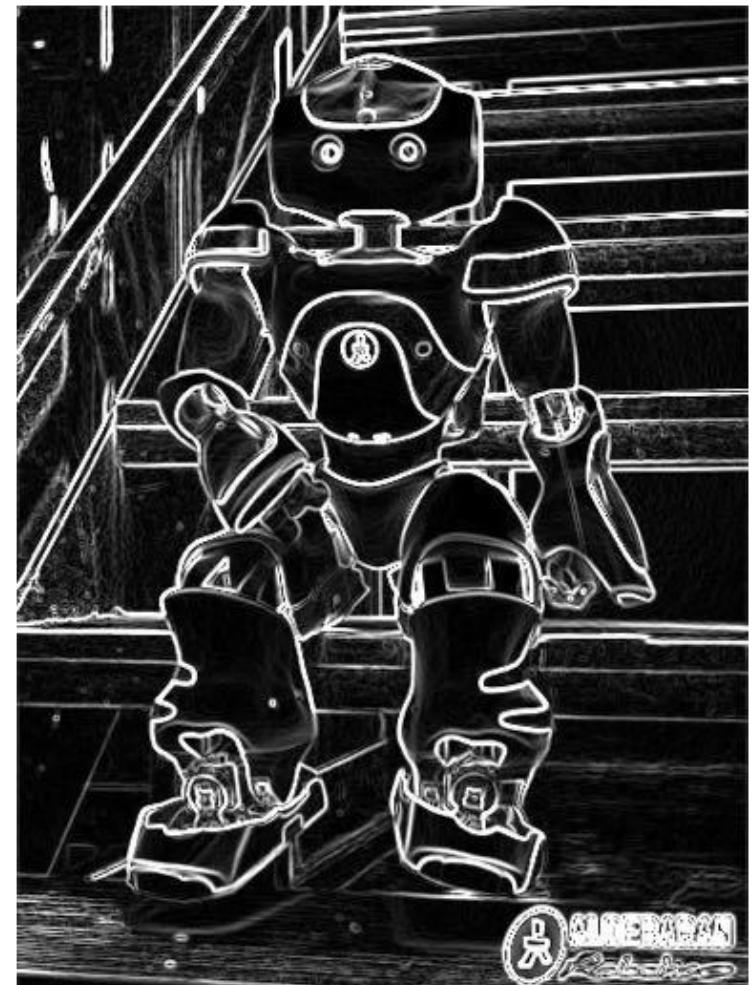
- En 2D: Filtre de Sobel

Dérivée selon x : I_x

$$H_1 = \frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Dérivée selon y : I_y

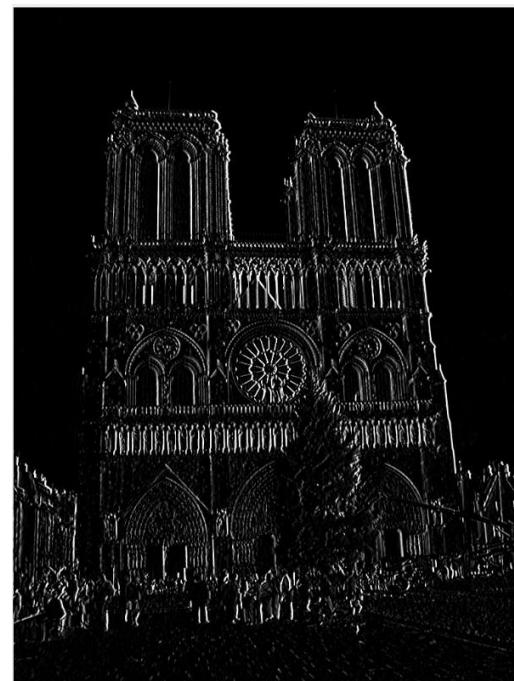
$$H_2 = \frac{1}{8} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$



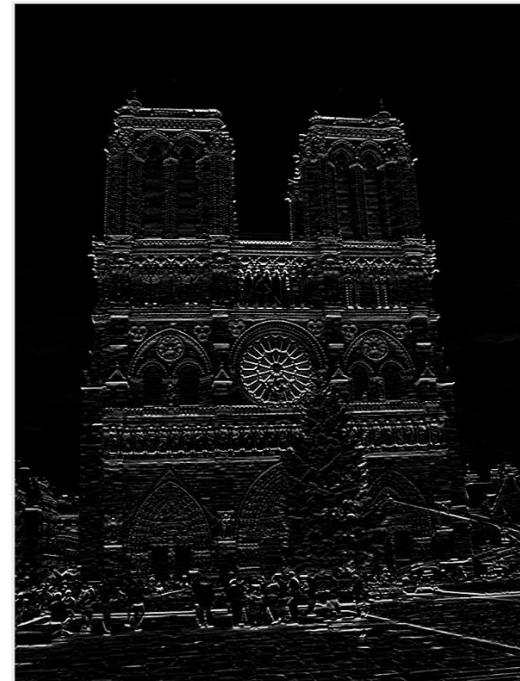
- Filtre de Sobel

Dérivée selon x : I_x

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$



Dérivée selon y : I_y



- Détection de contours
 - Application du Kernel:

$$\begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$



- Définition:
 - Zones où le gradient varie dans plusieurs directions
- DéTECTEUR de Harris:

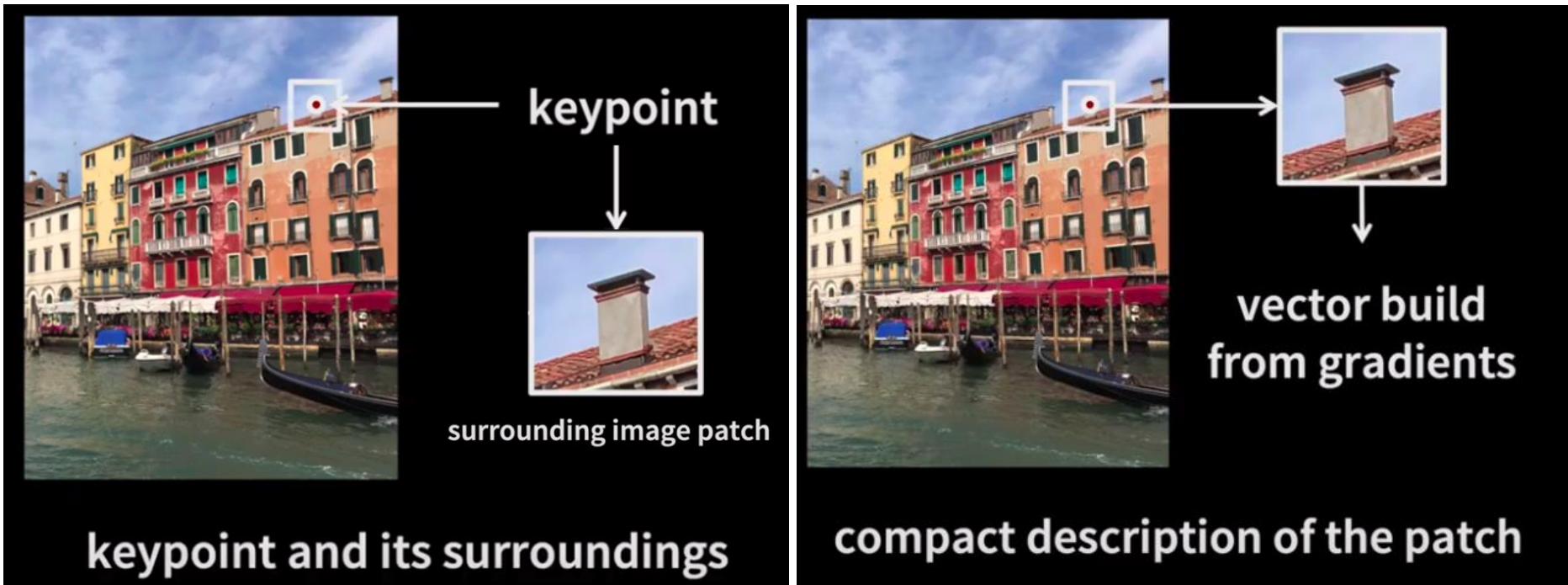


- Comment reconnaître un point après un changement d'échelle?

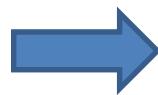


- Avec des descripteurs!

- Moyen de décrire une zone locale de l'image
- Les « features » sont associées à des points localement distincts dans l'image
- Les descripteurs sont la signature de ces points

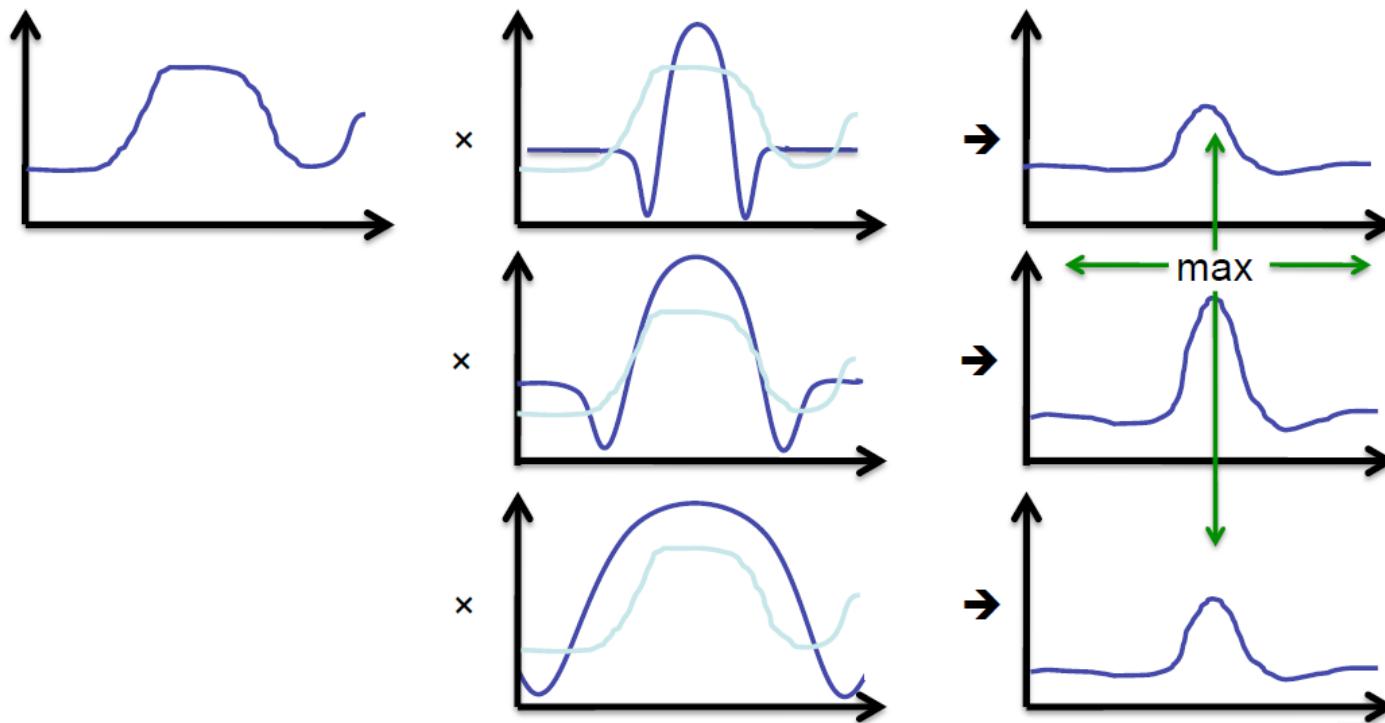


- Détection de blobs par différences de Gaussiennes:

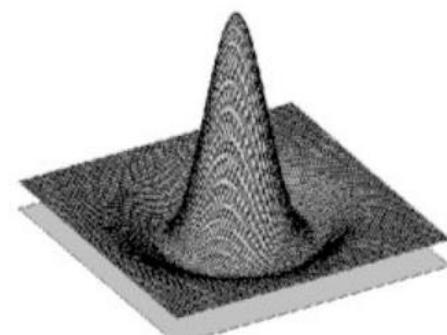
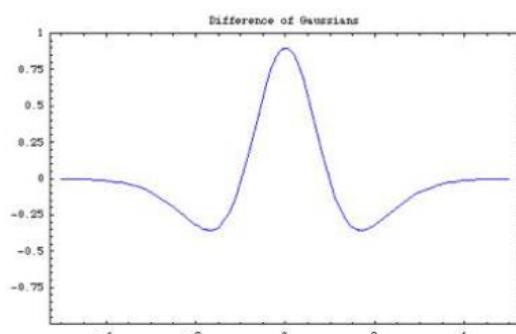


- Invariance par changement d'échelle pour les différences de Gaussiennes:

– Recherche de points au centre de « blobs »



- SIFT: Scale Invariant Feature Transform
 - Détection de blobs par la méthode des différences de Gaussiennes



Convolution

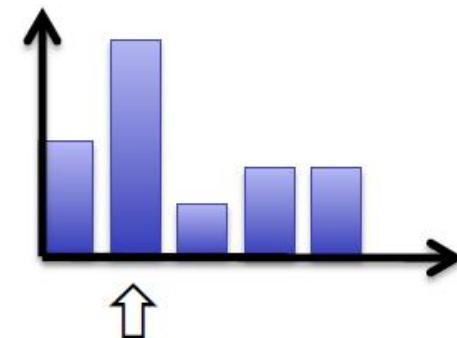
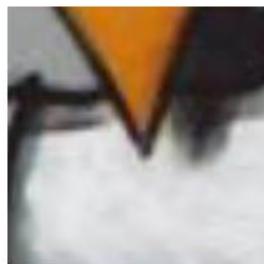


- Comment reconnaître un point après rotation?

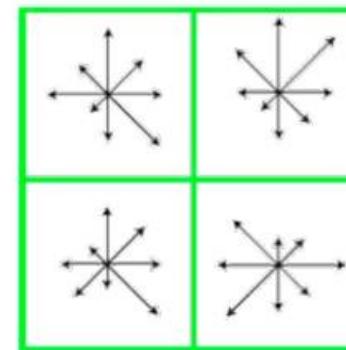
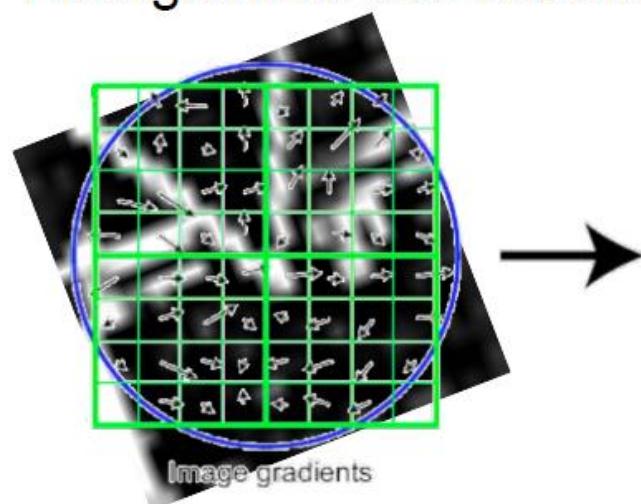


- Histogramme d'orientations du gradient

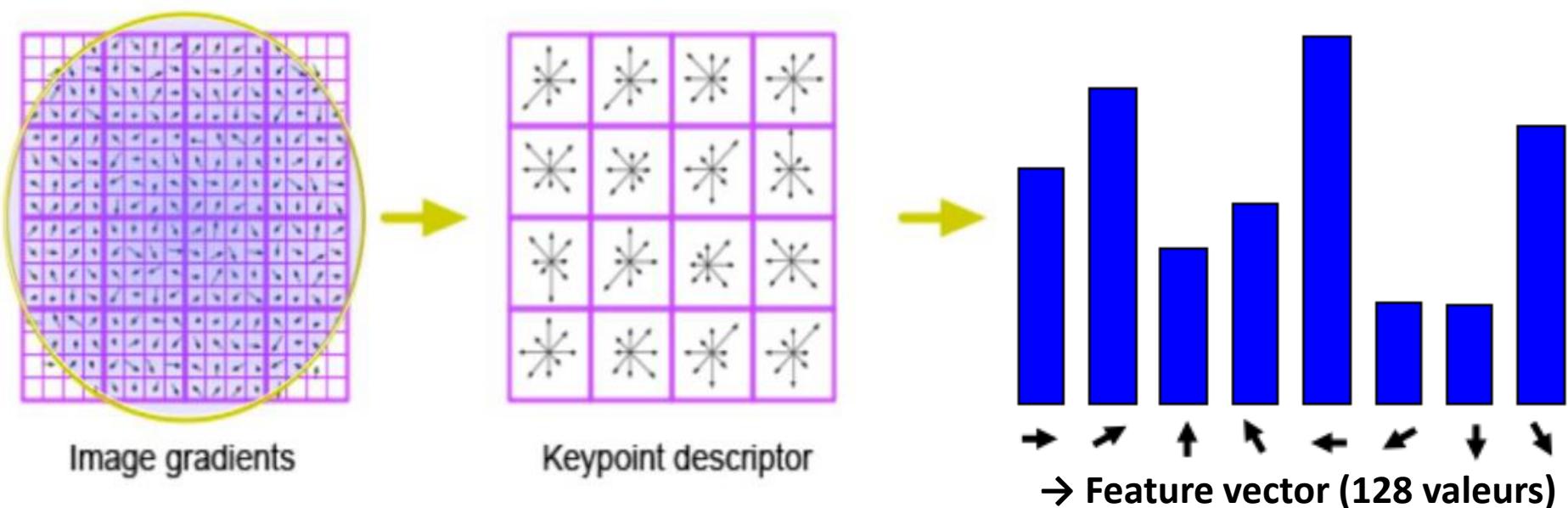
- Orientation de référence



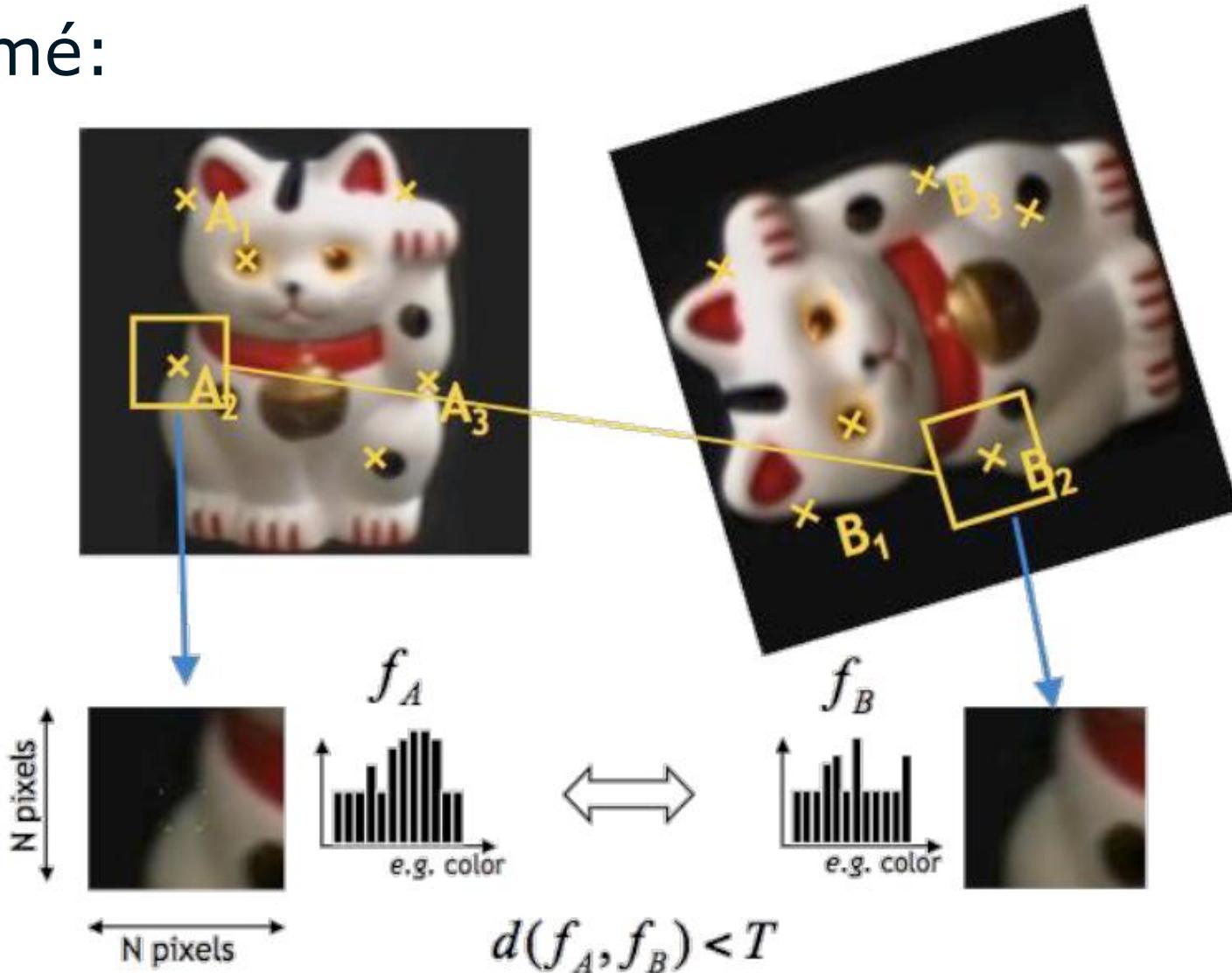
- Histogramme des orientations



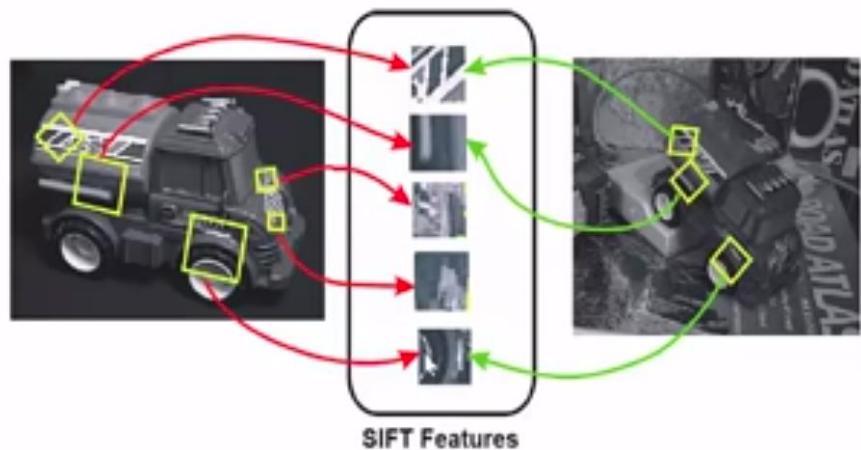
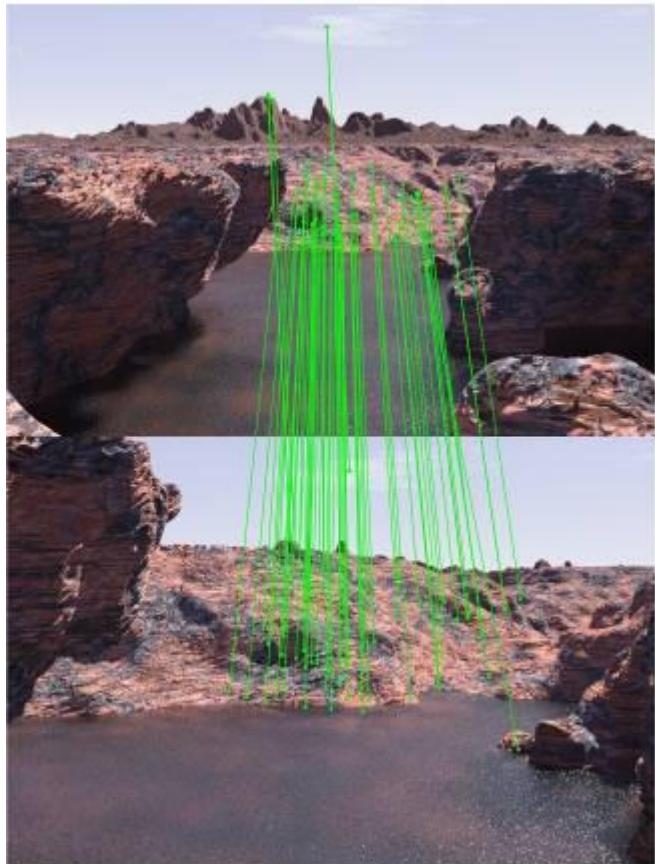
- Découpage en 4x4 fenêtres
- Histogramme sur 8 directions



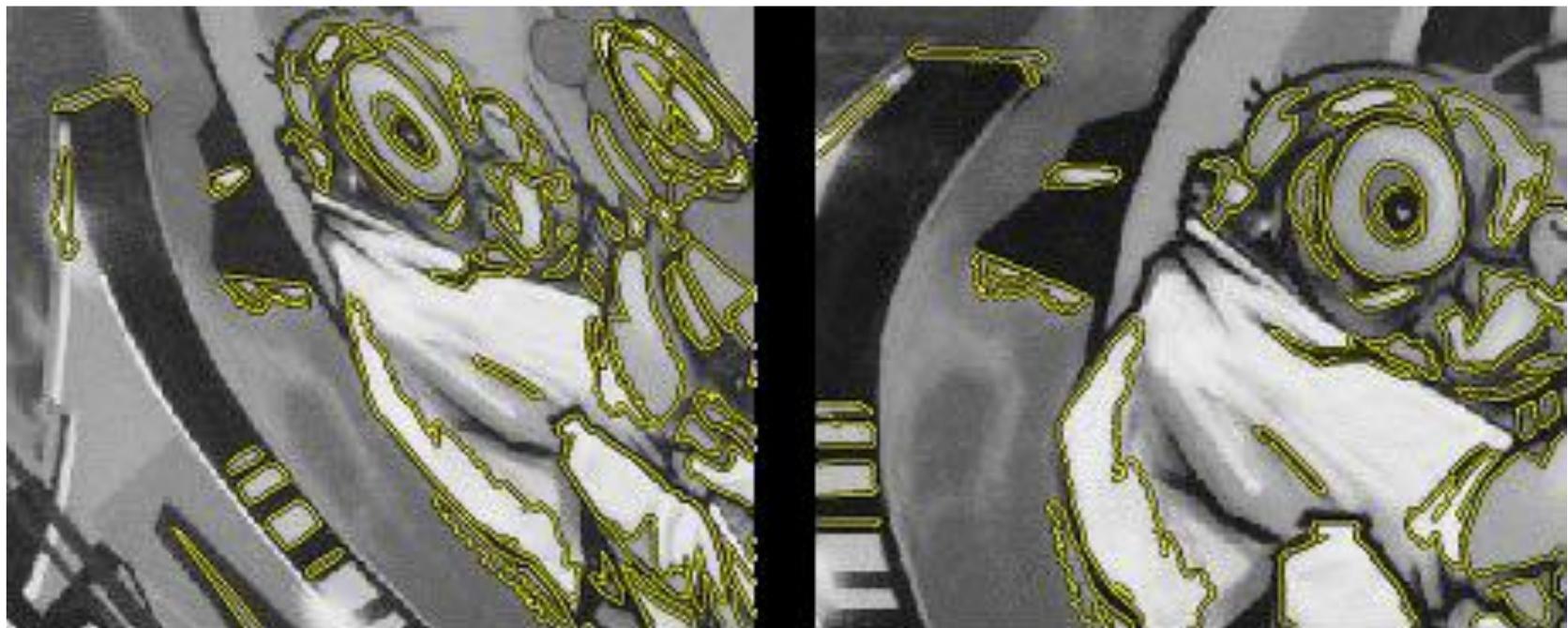
- Résumé:



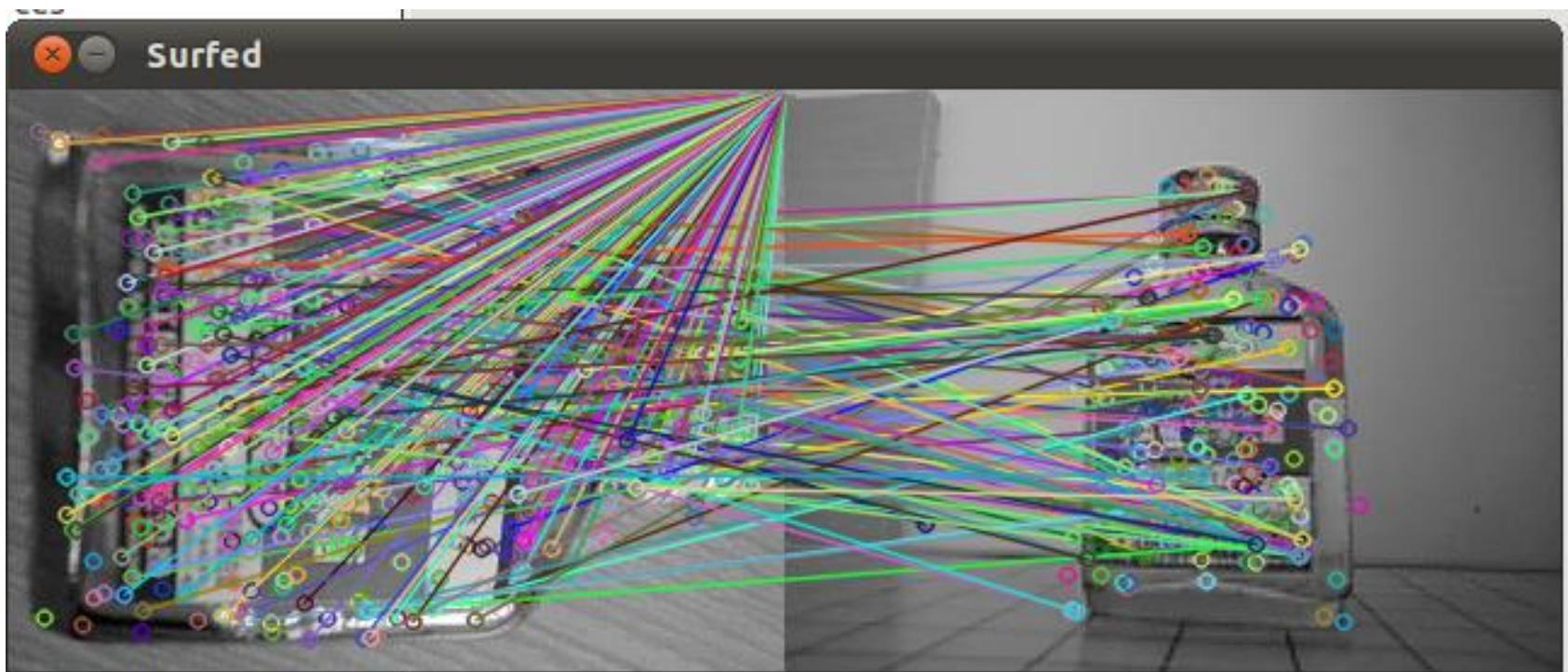
- Identification/Matching des keypoints



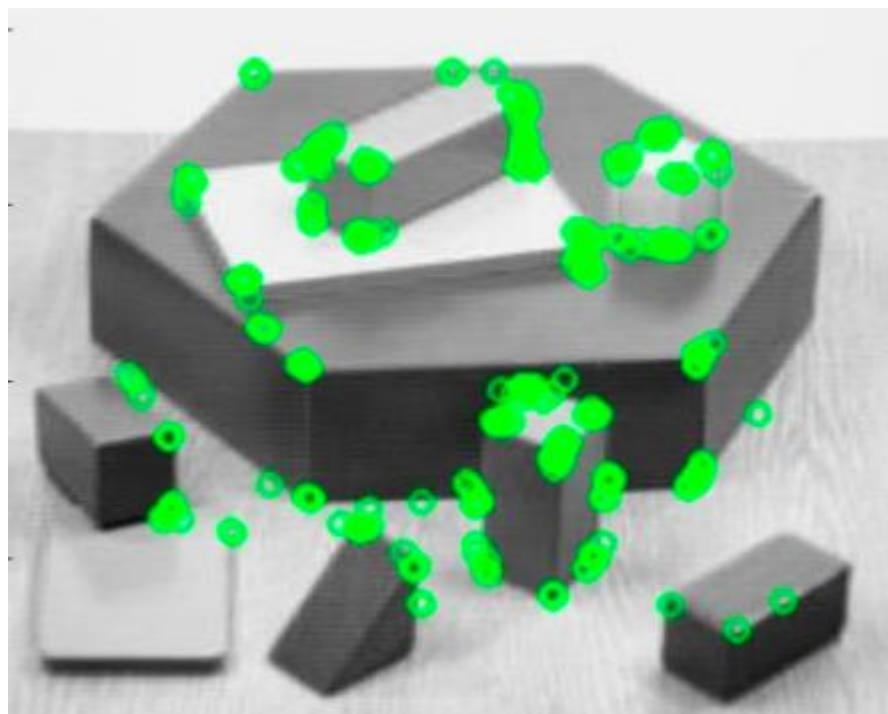
- MSER (Maximally Stable Extremal regions)
 - Segmentation de régions
 - Variation de paramètres
 - Conservation des régions stables



- SURF (Speeded Up Robust Features)
 - Inspiré de SIFT
 - Beaucoup plus rapide



- ORB (Oriented FAST and Rotated BRIEF)
 - SIFT et SURF sont brevetés: il faut payer si on veut les utiliser pour une application commerciale!
 - Le laboratoire d'OpenCV a inventé ORB comme alternative open-source et gratuite



- BRIEF(Binary Robust Independent Elementary Features)
- FAST(Features from Accelerated Segment Test)
- KAZE
- AKAZE
- BRISK
- ...



Extraction de caractéristiques locales

- Comment valoriser l'information?
 - Reconnaissance/détection d'objets
 - Estimation de la pose/localisation
 - De la projection d'objets 3D sur le plan image
 - D'objets 3D dans le monde
 - De la caméra dans le monde
 - Estimation du mouvement

Partie III: Détection et Localisation

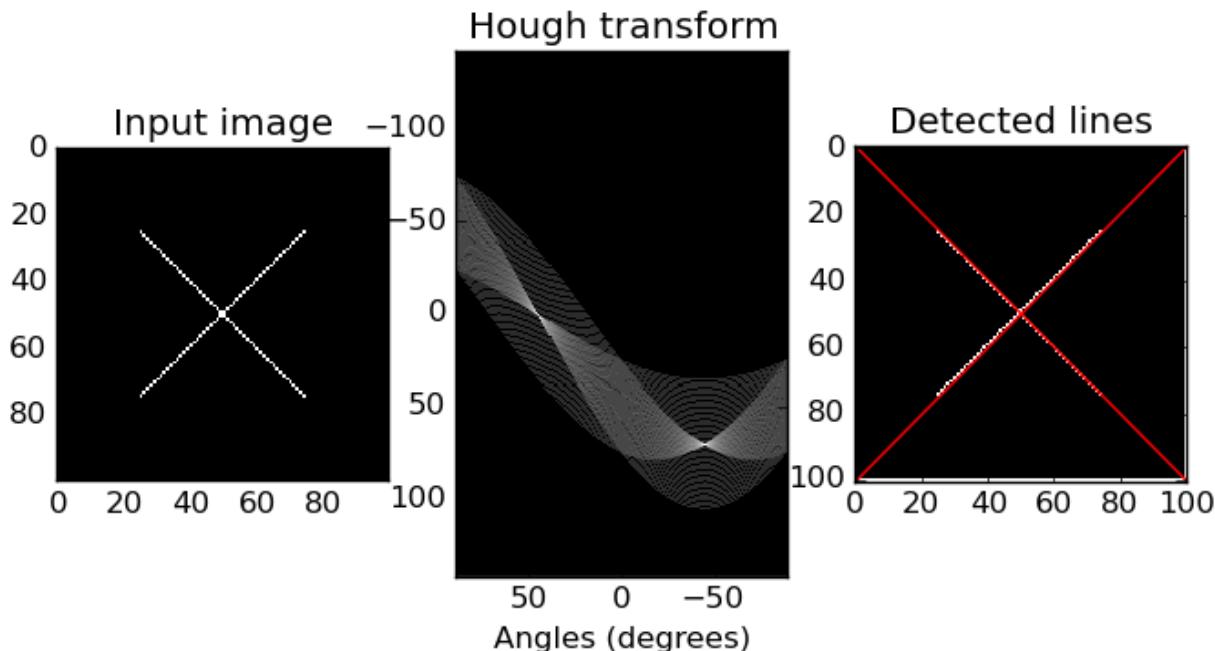
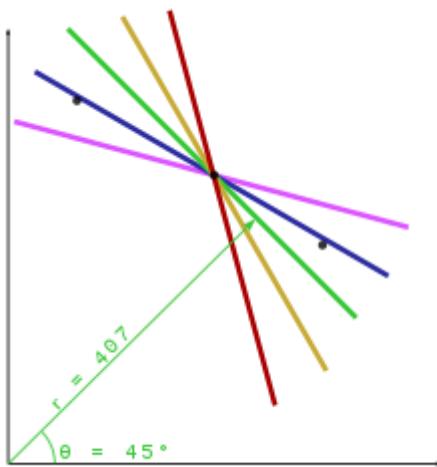
- I. Introduction
- II. Géométrie projective
- III.Extraction de caractéristiques locales
- IV. Reconnaissance d'objets
- V. Estimation du mouvement

- Objectifs:

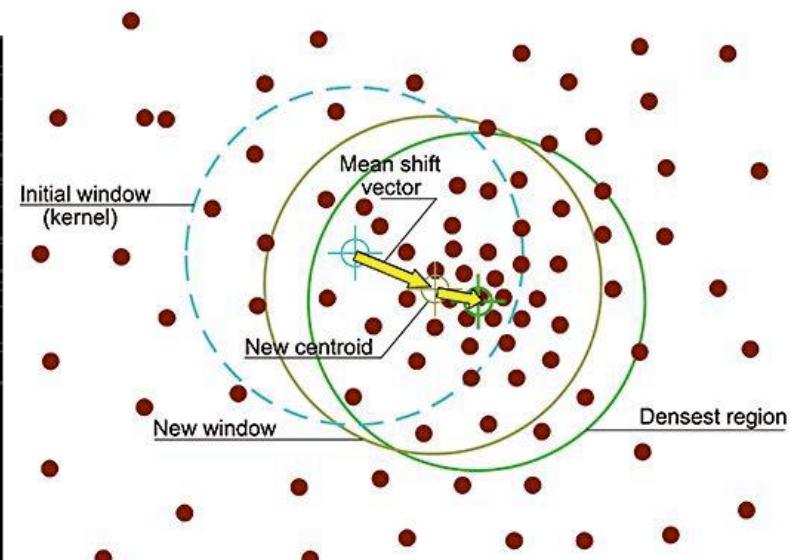
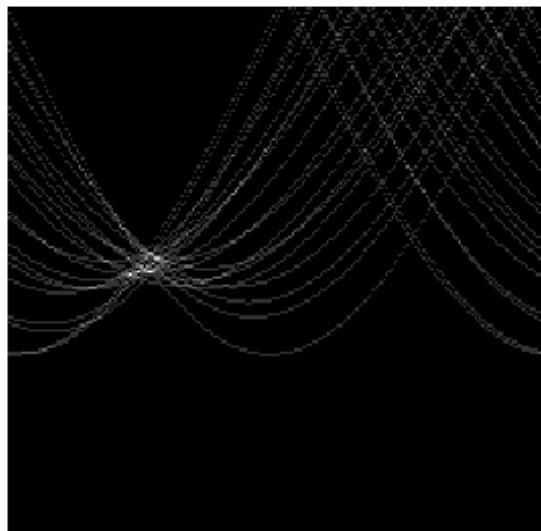
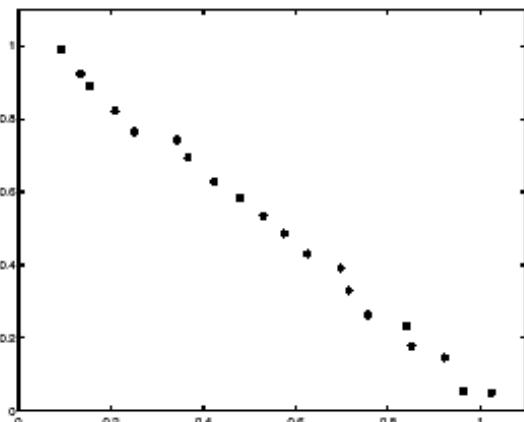
- Détection d'instances d'objets par points d'intérêt
 - Transformée de Hough
 - RANSAC
- Détection de catégories d'objets
 - Modèles de Sac de mots visuels



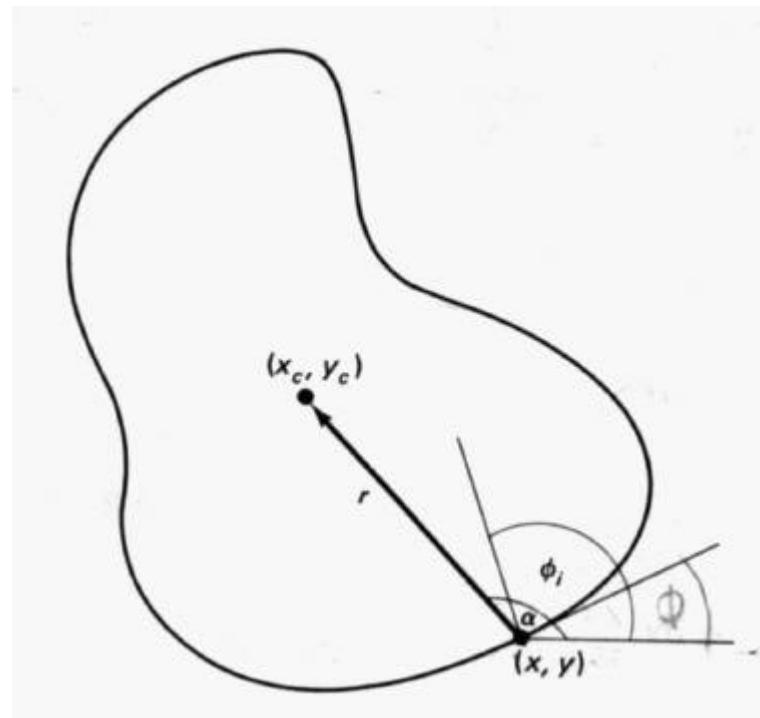
- À l'origine, détection de lignes droites:
 - Chaque point vote pour « toutes » les lignes qui passent par lui
 - Les votes sont accumulés
 - Un maximum local correspond à des lignes candidates



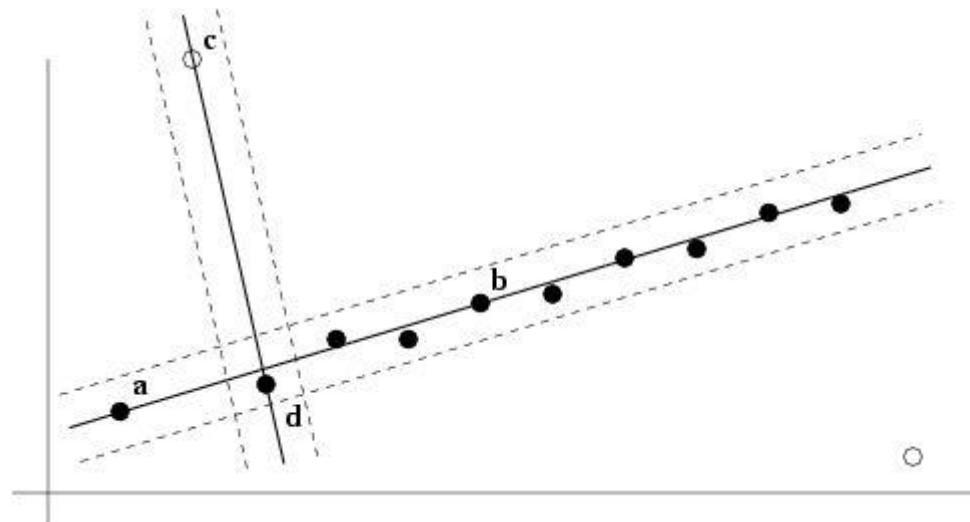
- Possible problème: Trouver le maximum vrai
 - Mean shift
 - Gaussian convolution
 - ...



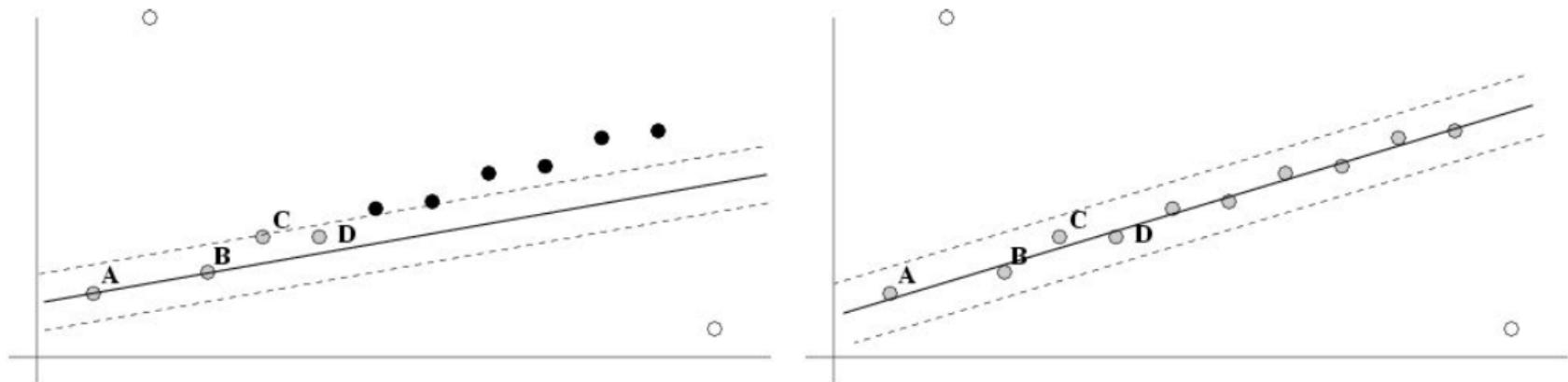
- Transformée de Hough généralisée
 - Contour/forme arbitraire
 - Choix d'un point de référence pour le contour (e.g. centre)
 - Pour chaque point du contour, se rappeler de sa position par rapport au point de référence
 - Calcul de l'angle tangente en chaque point et vote pour tous les points de référence possibles ou toutes les transformations possibles



- RANSAC (RANdom Sample Consensus)
 - Cas de lignes
 - Choix aléatoire de droites
 - Vote basé sur le nombre de points proches de la ligne
 - La ligne qui a le plus de votes est choisie



- Amélioration
 - Élimination des outliers par RANSAC
 - Amélioration de l'estimation de RANSAC par un autre algorithme (e.g. minimisation moindres carrés, ICP en 3D, etc.)



• Comparaison

Gen. Hough Transform

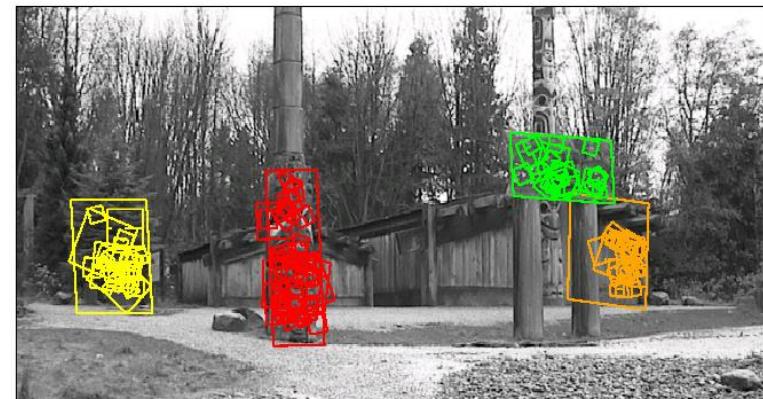
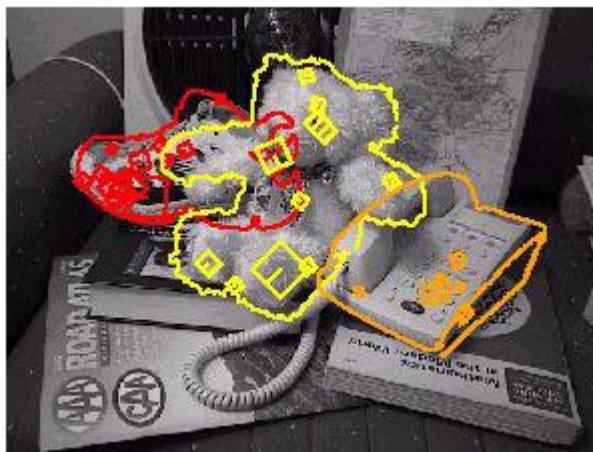
- Advantages
 - Very effective for recognizing arbitrary shapes or objects
 - Can handle high percentage of outliers (>95%)
 - Extracts groupings from clutter in linear time
- Disadvantages
 - Quantization issues
 - Only practical for small number of dimensions (up to 4)
- Improvements available
 - Probabilistic Extensions
 - Continuous Voting Space

} [Leibe08]

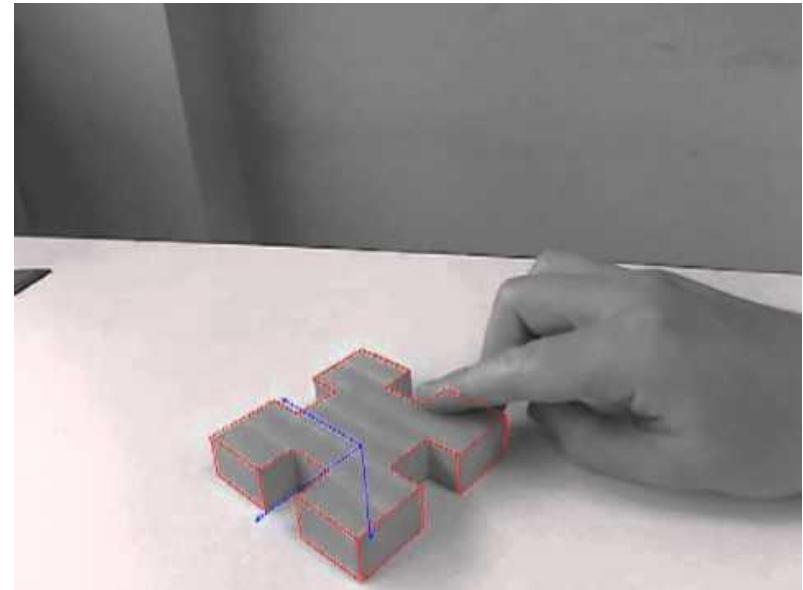
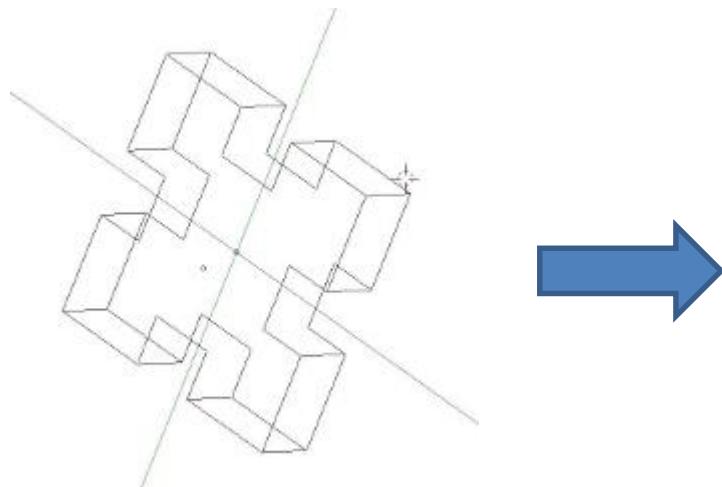
RANSAC

- Advantages
 - General method suited to large range of problems
 - Easy to implement
 - Independent of number of dimensions
- Disadvantages
 - Only handles moderate number of outliers (<50%)
- Many variants available, e.g.
 - PROSAC: Progressive RANSAC [Chum05]
 - Preemptive RANSAC [Nister05]

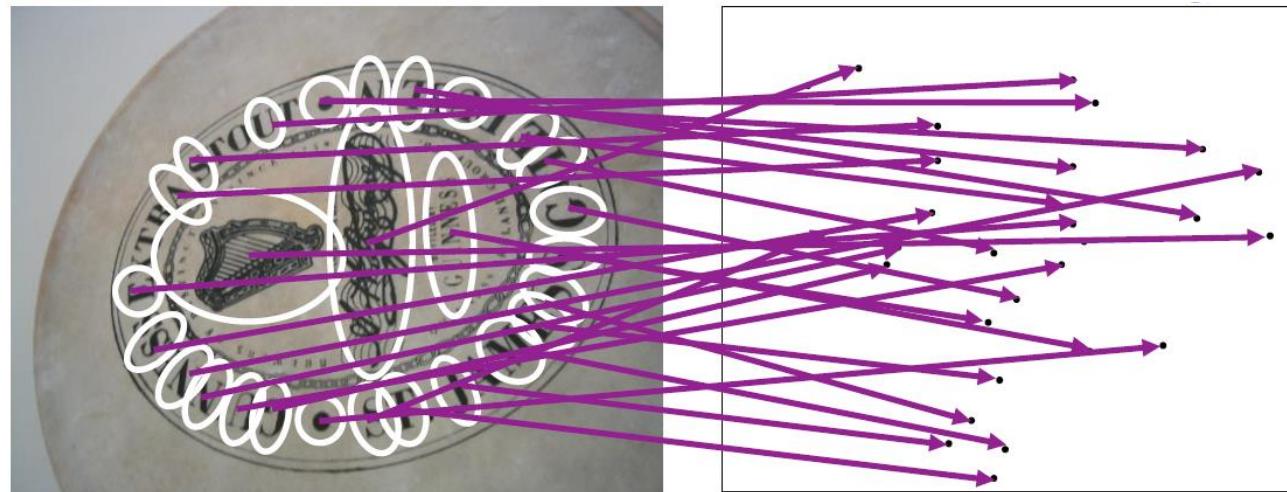
- Reconnaissance d'objets 3D
 - Basé sur la détection de features:
 - 3 features minimum sont nécessaires pour la reconnaissance



- Reconnaissance d'objets 3D basée sur la détection d'un modèle 3D connu

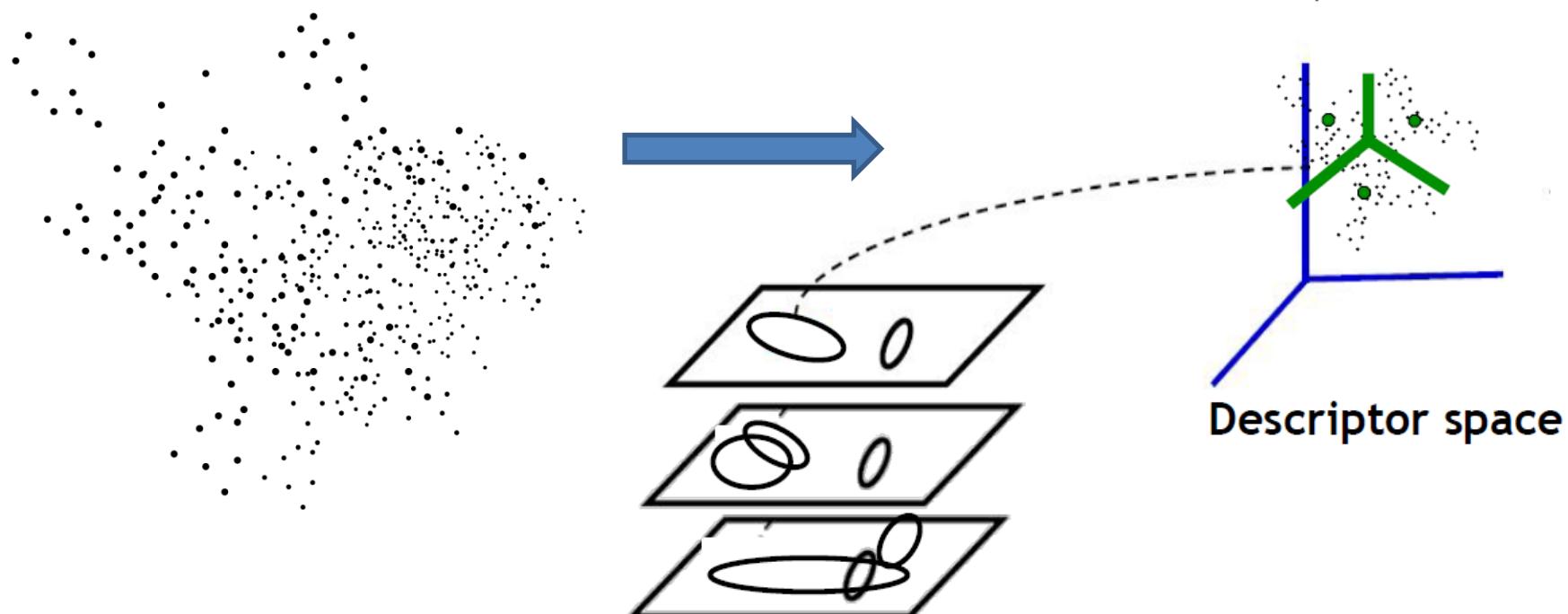


- Principe
 - Extraction de features locales à partir d'un certain nombre d'images



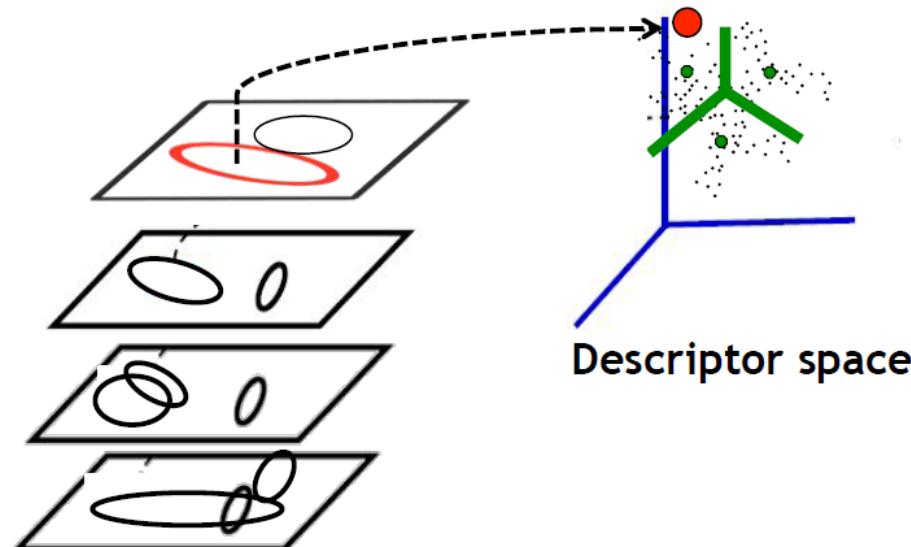
- ## Principe

- Cartographie des descripteurs vers de mots visuels qui quantifient l'espace des features
- Le centre des clusters définissent les prototypes de mots

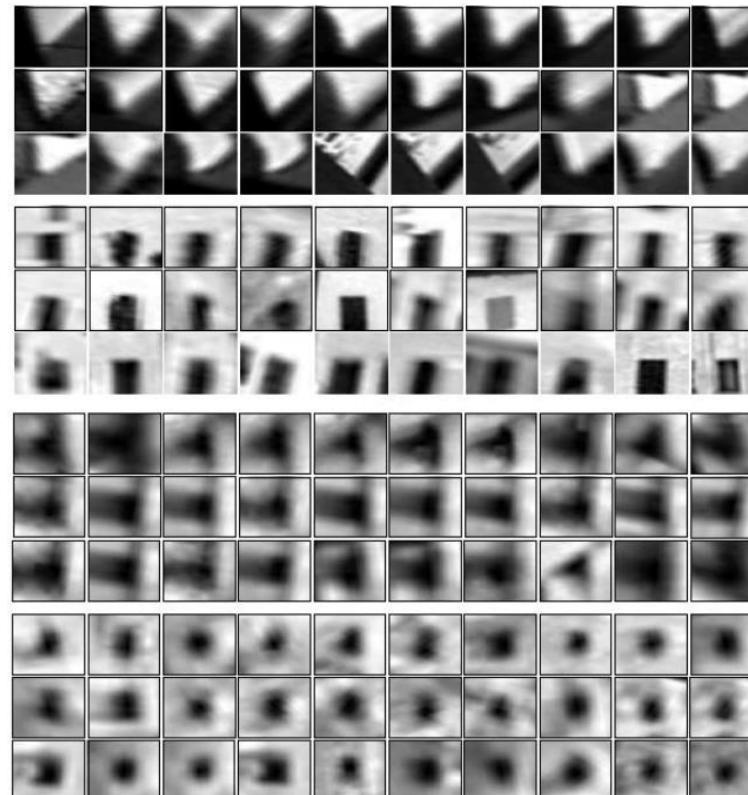


- **Principe**

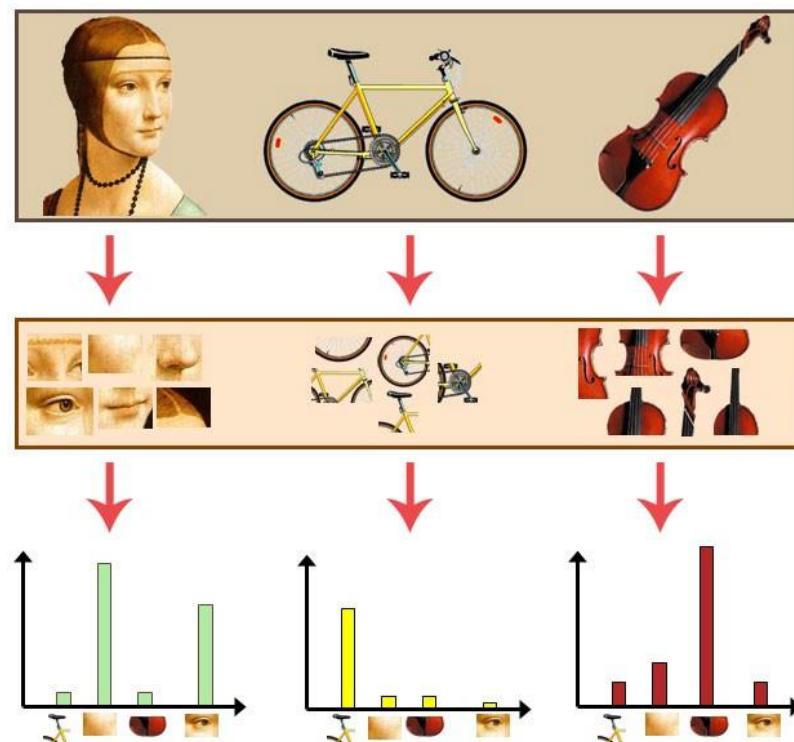
- Cartographie des descripteurs vers de mots visuels qui quantifient l'espace des features
- Détermination de quel mot doit être assigné à chaque nouvelle région de l'image en trouvant le centre du cluster le plus proche



- Exemple:
 - Chaque groupe de patch correspond à un même mot visuel



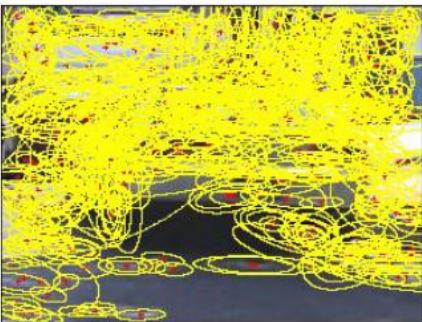
- Résumer une image entière à partir de sa distribution de présence de mots
- Analogue à un sac de mots souvent utilisé pour les documents texte



- **Création d'un Vocabulaire visuel:**
 - Répertorier un ensemble de mots visuels (~dictionnaire)
 - Différentes stratégies:
 - Apprentissage supervisé ou non
 - Clustering
 - Échantillonage de descripteurs
 - Deep-Learning
 - ...



- Stratégies d'échantillonnage:



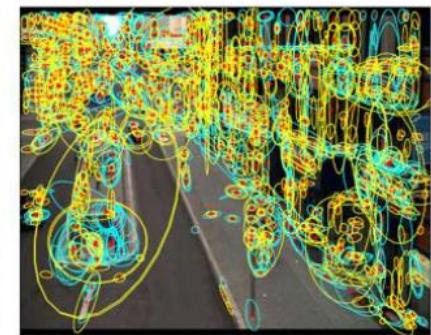
Sparse, at
interest points



Dense, uniformly

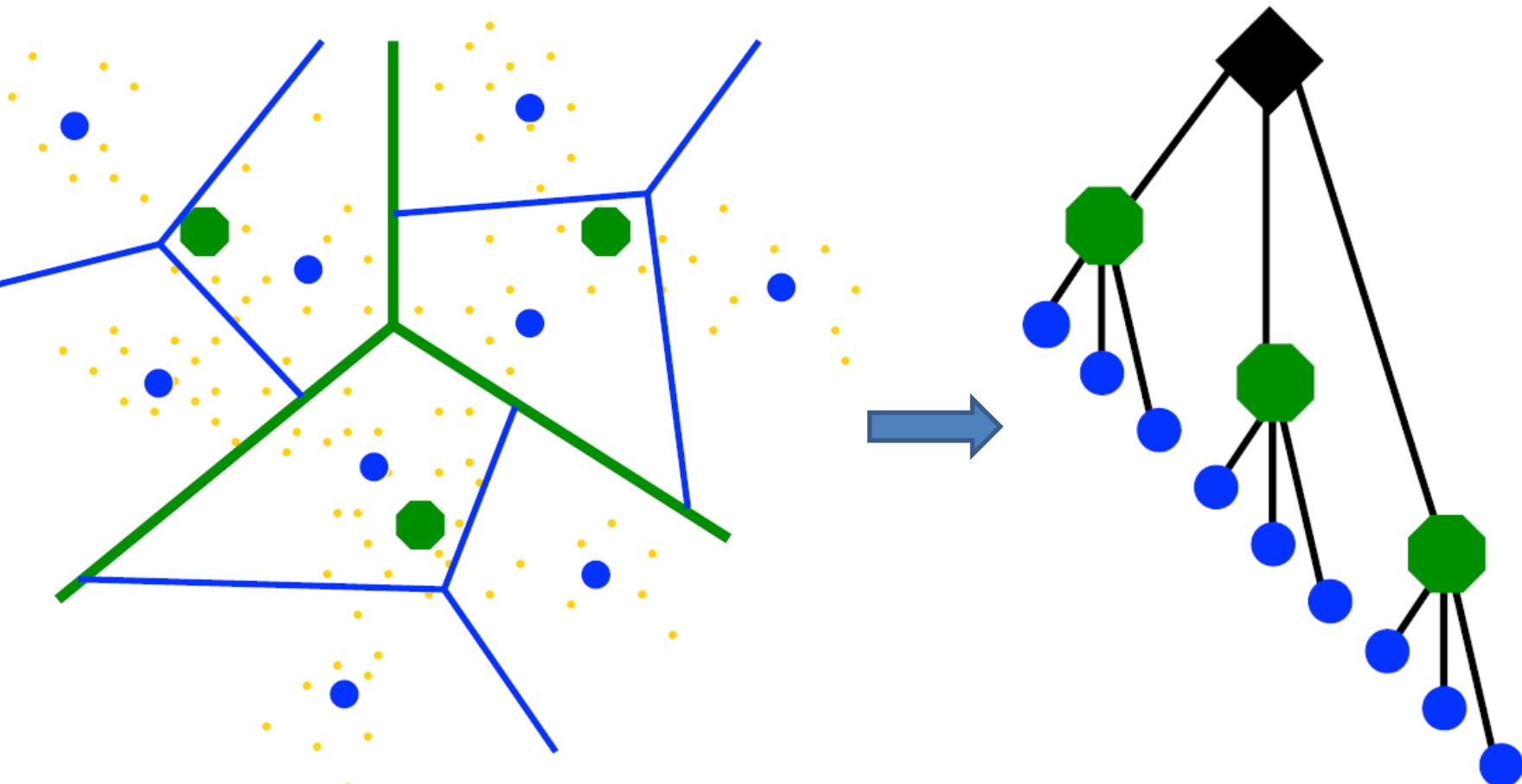


Randomly

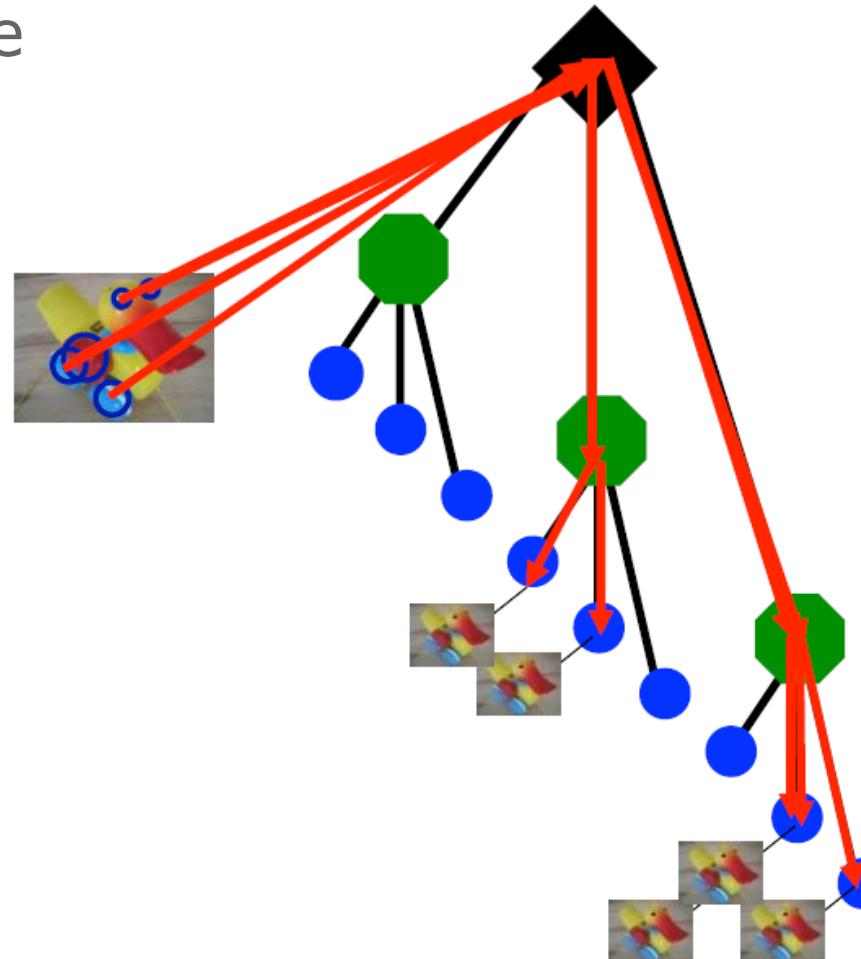


Multiple interest
operators

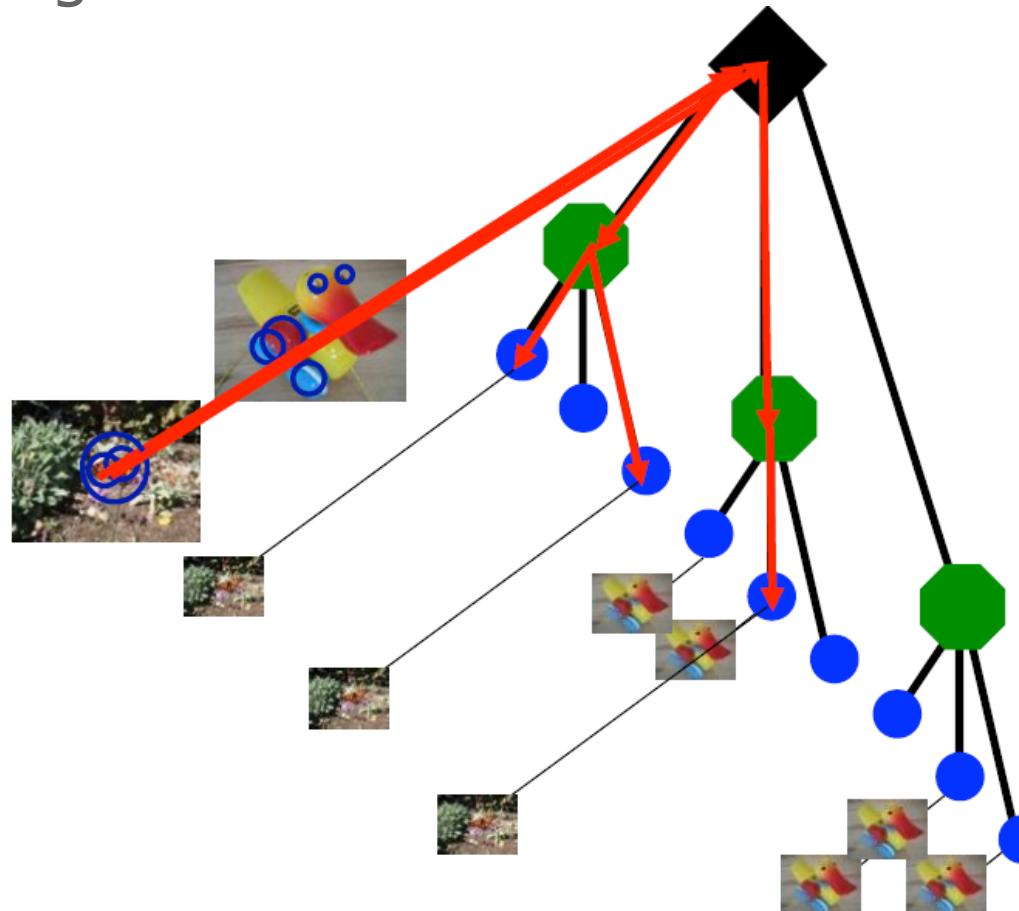
- Arbre de vocabulaire:



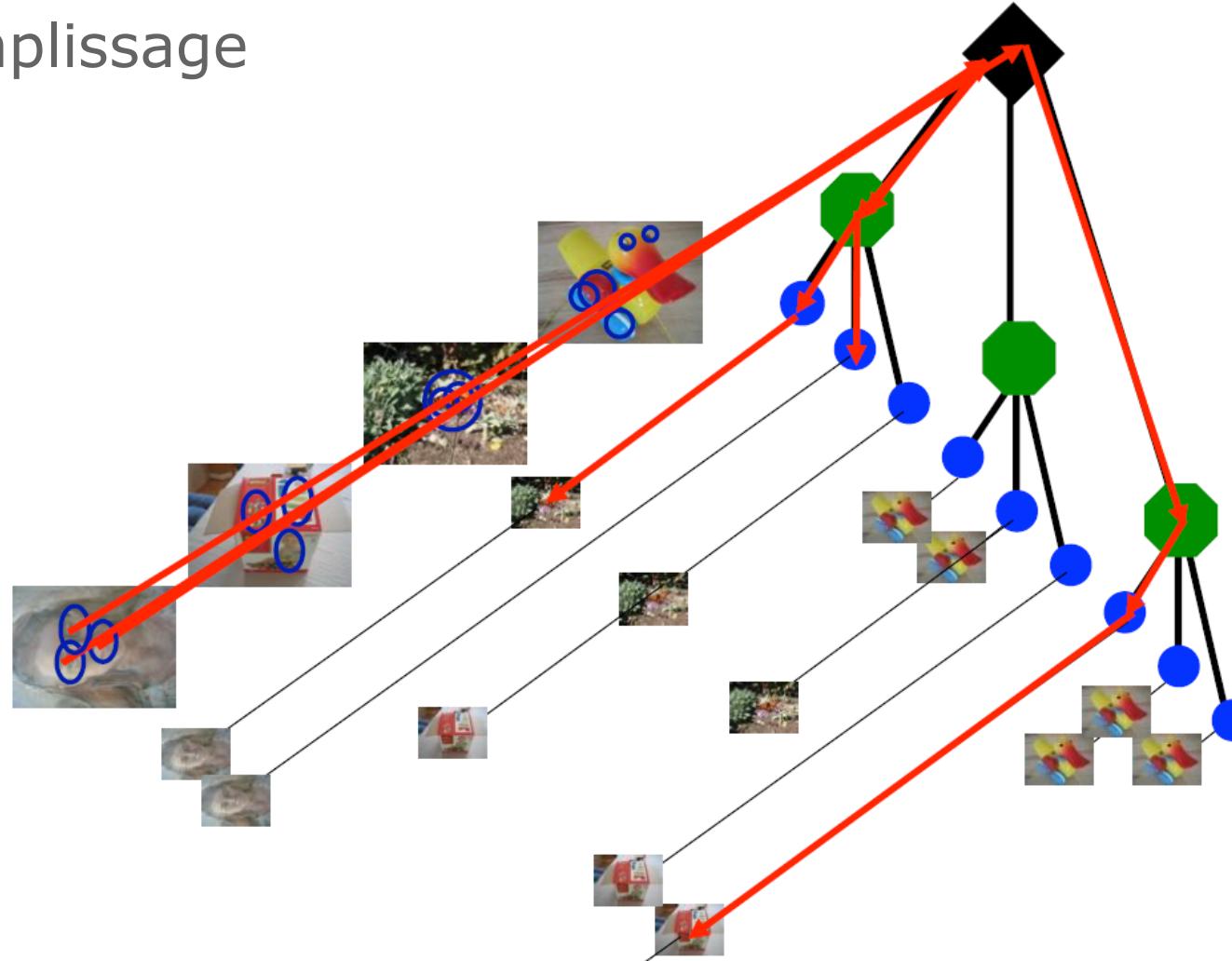
- Arbre de vocabulaire:
 - Remplissage



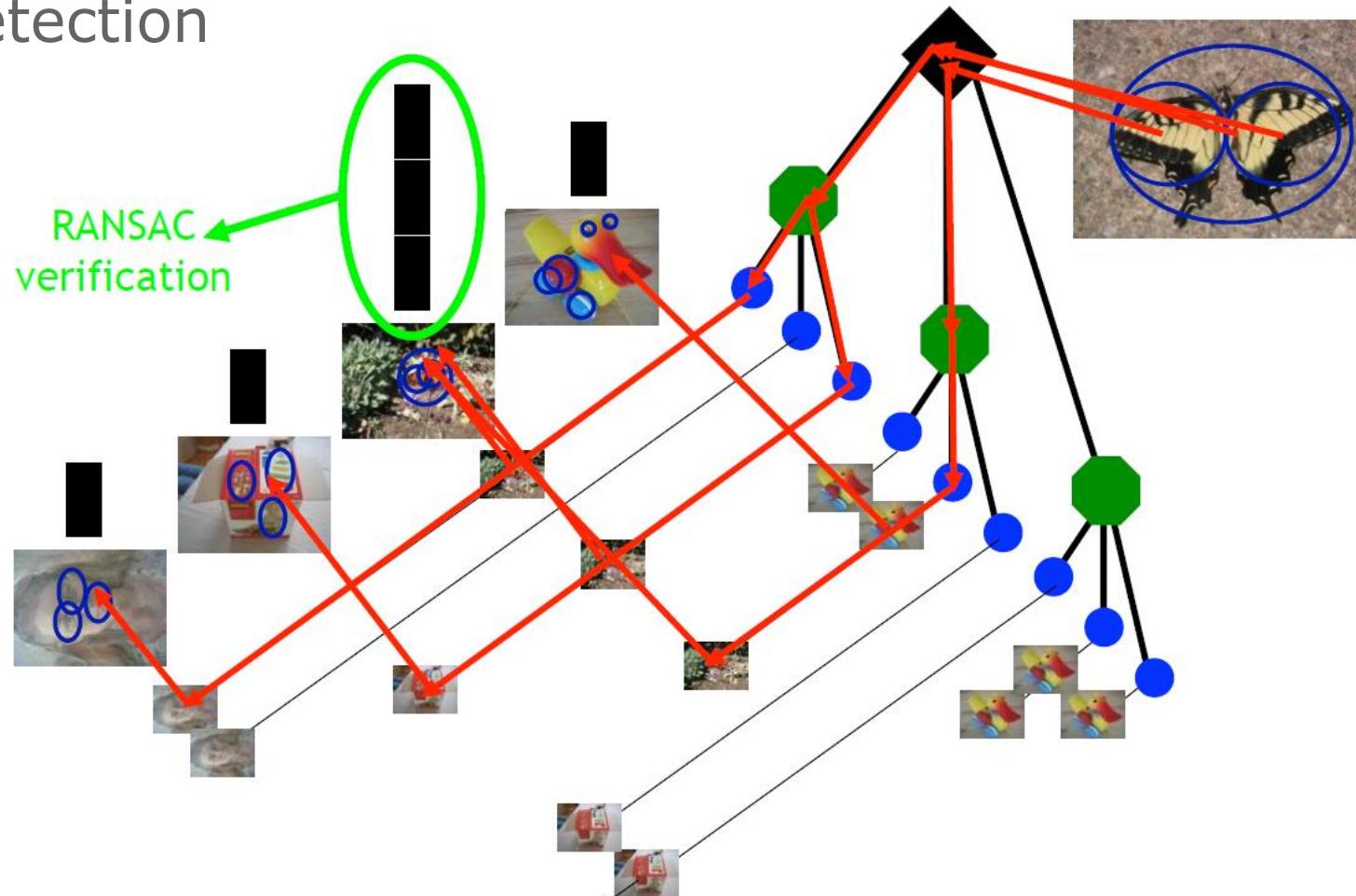
- Arbre de vocabulaire:
 - Remplissage



- Arbre de vocabulaire:
 - Remplissage



- Arbre de vocabulaire:
 - Détection



- **Problème:**
 - Certain mots visuels sont discriminants
 - D'autres apparaissent dans de nombreuses images
- **Calcul d'un poids pour chaque mot visuel**
 - Le poids corresponds à la quantité espérée d'information que ce mot apporte
 - Normalisation des histogrammes en fonction de ces poids

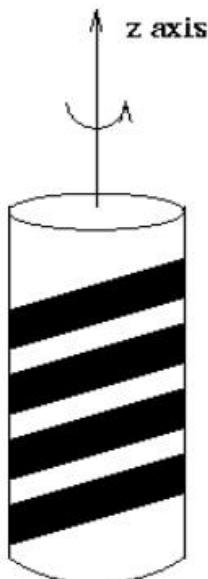
Partie III: Détection et Localisation

- I. Introduction
- II. Géométrie projective
- III. Extraction de caractéristiques locales
- IV. Reconnaissance d'objets
- V. Estimation du mouvement

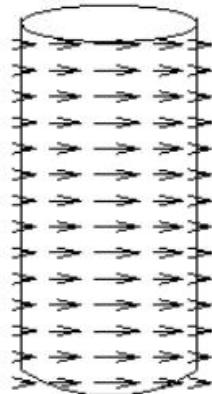
- Objectifs:

- Détection/Estimation du mouvement dans la scène
 - Dû au mouvement de la caméra
 - Mouvement des objets
- Perception du mouvement apparent
 - Champ des vecteurs de déplacement
 - Flux optique

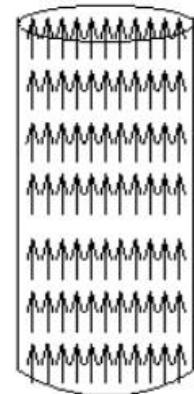
- Difficultés de l'estimation du flux optique
 - Ambiguïtés



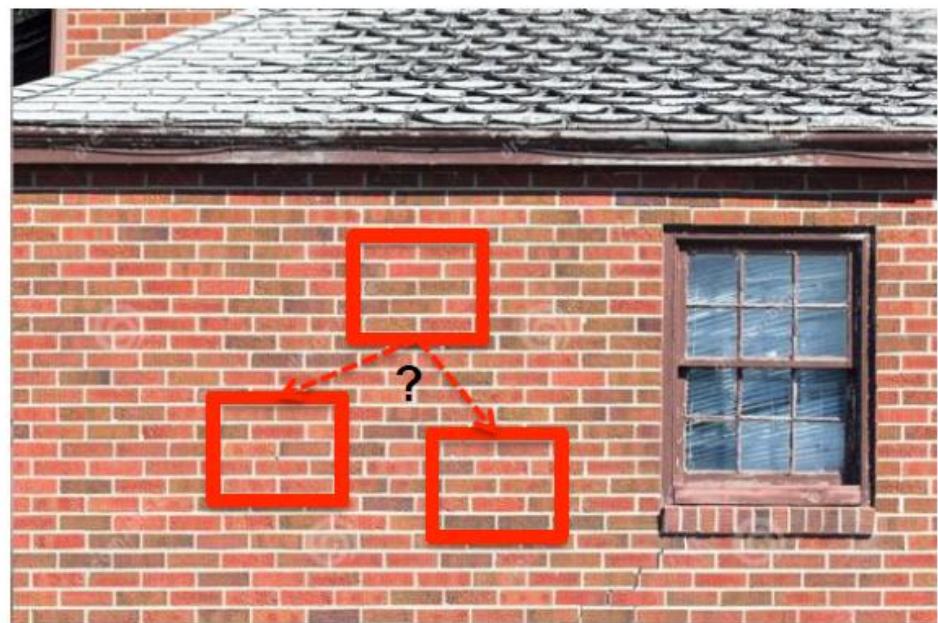
Barber's pole



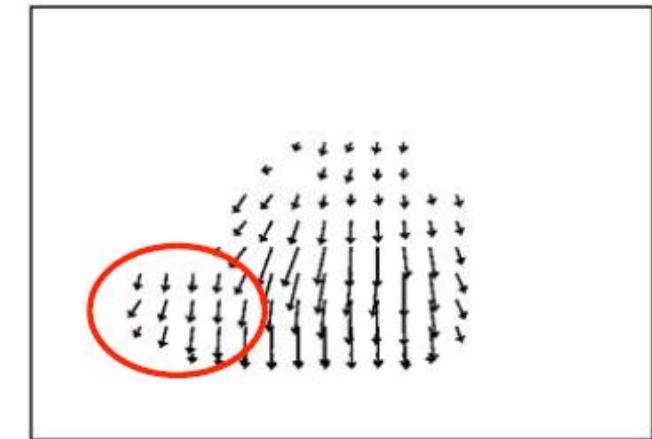
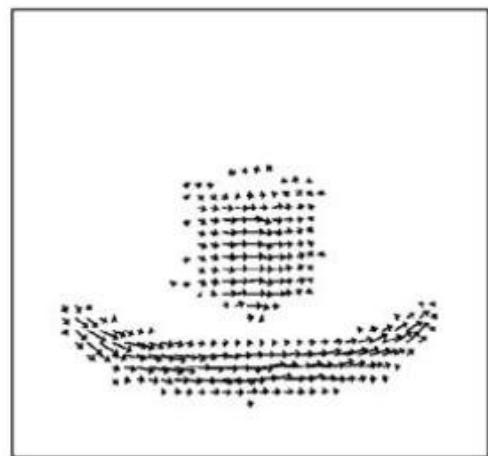
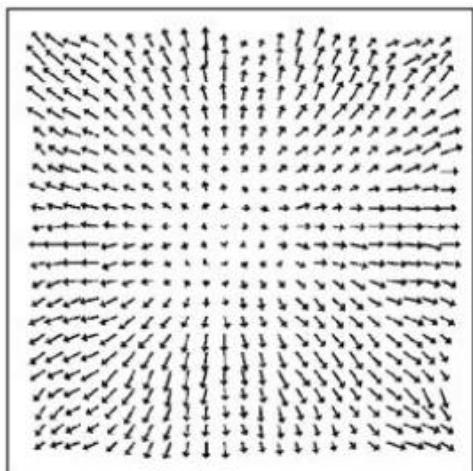
Motion field



Optical flow

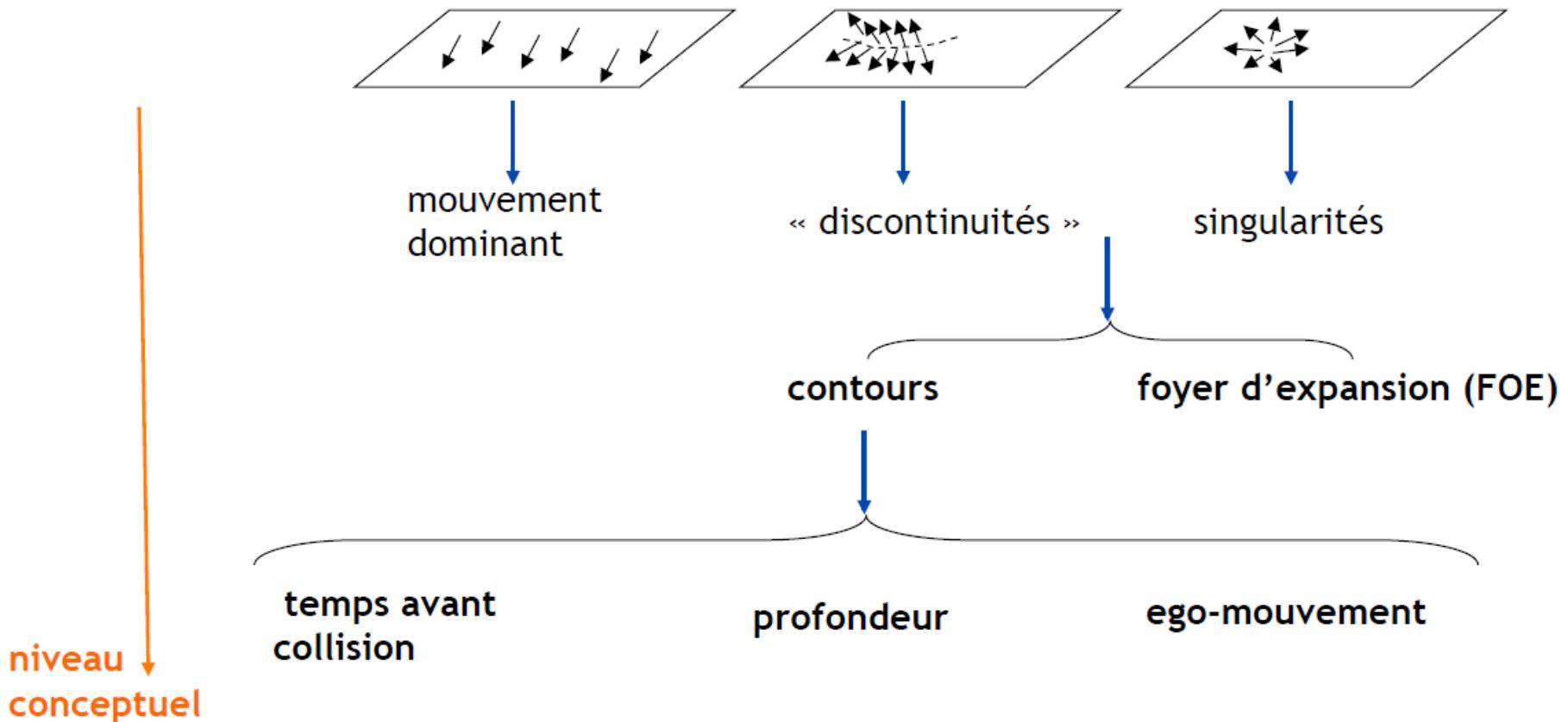


- Exemples:



Source : Pierre Kornprobst - INRIA

- Interprétation du flux:



- Vitesse:

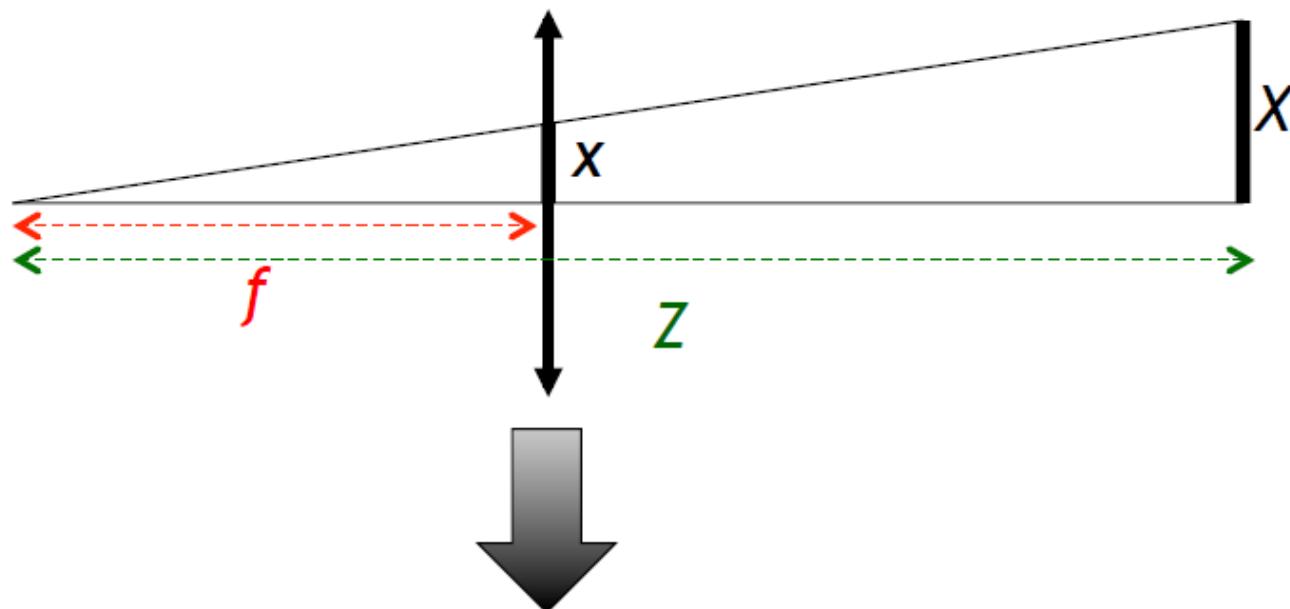
- La caméra se déplace à une vitesse (X', Y', Z') par rapport à la scène.
- Si on dérive les équations de perspective on a donc:

$$\bullet \quad \begin{cases} u' = f_x \left(\frac{X'}{Z} - \frac{XZ'}{Z^2} \right) \\ v' = f_y \left(\frac{Y'}{Z} - \frac{YZ'}{Z^2} \right) \end{cases}, \text{ avec } \begin{cases} X' = \frac{\partial X}{\partial t} \\ Y' = \frac{\partial Y}{\partial t} \\ Z' = \frac{\partial Z}{\partial t} \end{cases}$$

- Translation pure selon X (ou Y):

- $Z' = 0$ et $Y' = 0$ (resp. $X' = 0$)

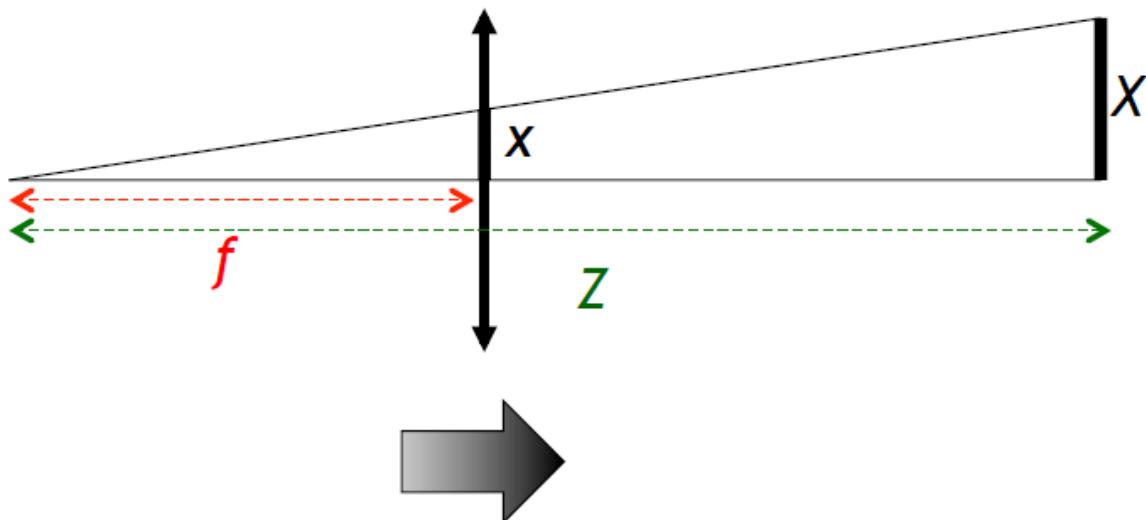
- $\begin{cases} x' = f_x \frac{x'}{z} \\ y' = 0 \end{cases}$ donc $x' \sim \frac{1}{z}$



- Translation pure selon Z:

– $\begin{cases} X' = 0 \\ Y' = 0 \end{cases}$, donc $\begin{cases} u' = f_x \frac{XZ'}{Z^2} = \frac{(u - c_x)Z'}{Z} \\ v' = f_y \frac{YZ'}{Z^2} = \frac{(v - c_y)Z'}{Z} \end{cases}$

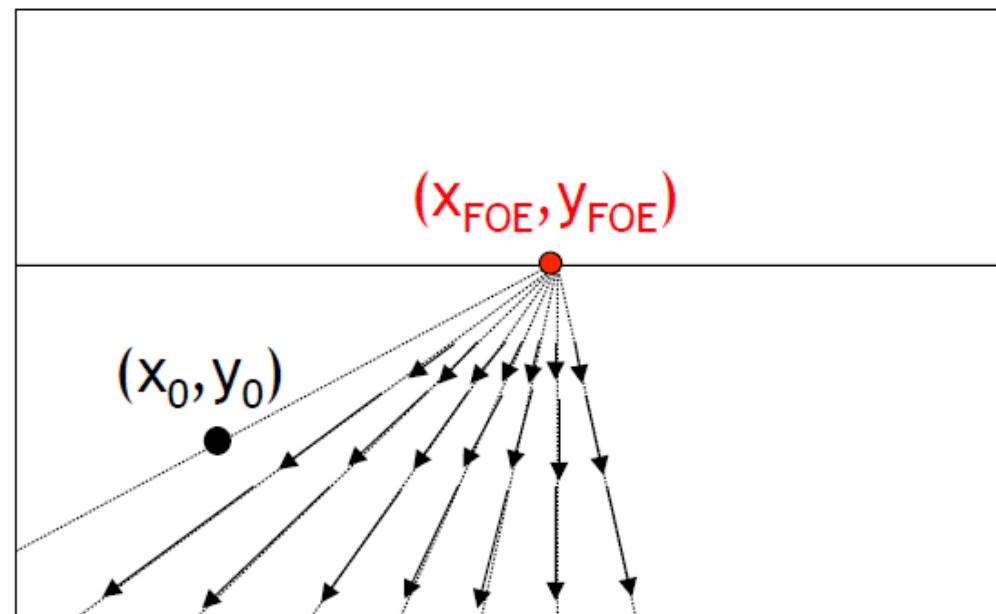
– Foyer d'expansion au niveau du centre optique



- Cas général:
 - Donne la direction du déplacement
 - Mouvement (X' , Y' , Z')
 - Soit $[X_0, Y_0, Z_0]^T$ un point de la scène, après un temps t , il est projeté sur l'image au point $[u_t, v_t]^T$, avec:

$$\bullet \quad \begin{cases} u_t = f_x \frac{X_0 + tX'}{Z_0 + tZ'} \\ v_t = f_x \frac{Y_0 + tY'}{Z_0 + tZ'} \end{cases}$$

$$\bullet \quad \begin{cases} u_{FOE} = \lim_{t \rightarrow -\infty} u_t \\ v_{FOE} = \lim_{t \rightarrow -\infty} v_t \end{cases}$$



- Temps avant collision:

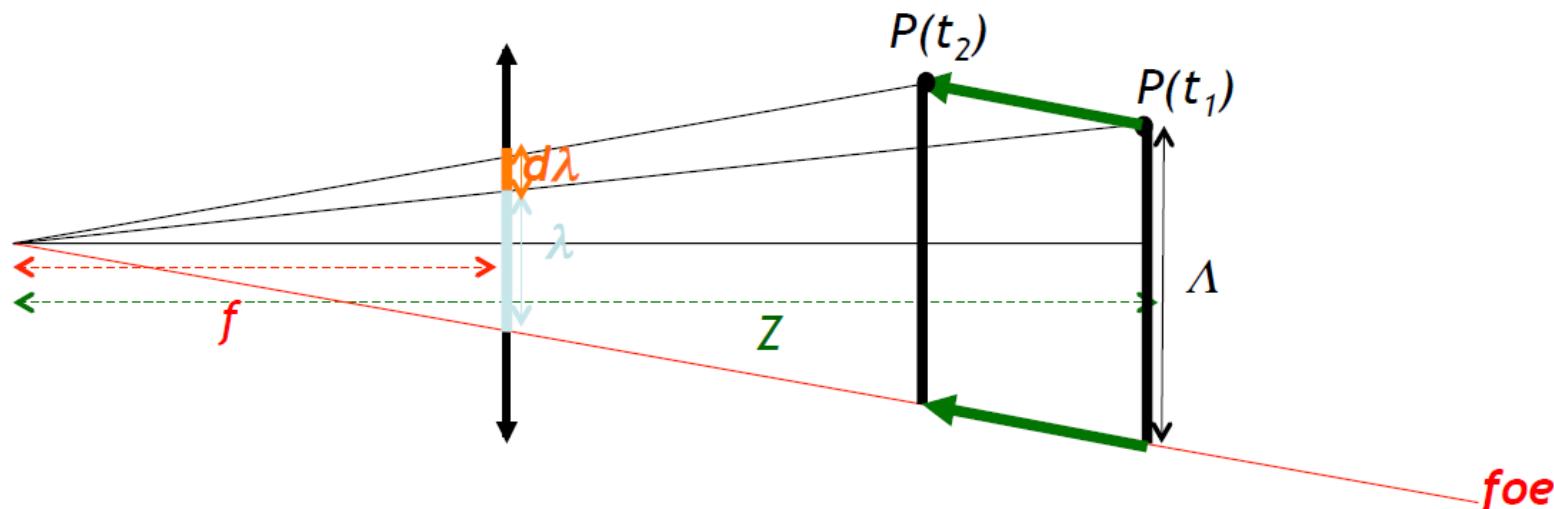
- Mesure de la taille d'un élément: $\lambda = f \frac{\Lambda}{z}$

- Mouvement dans une scène statique:

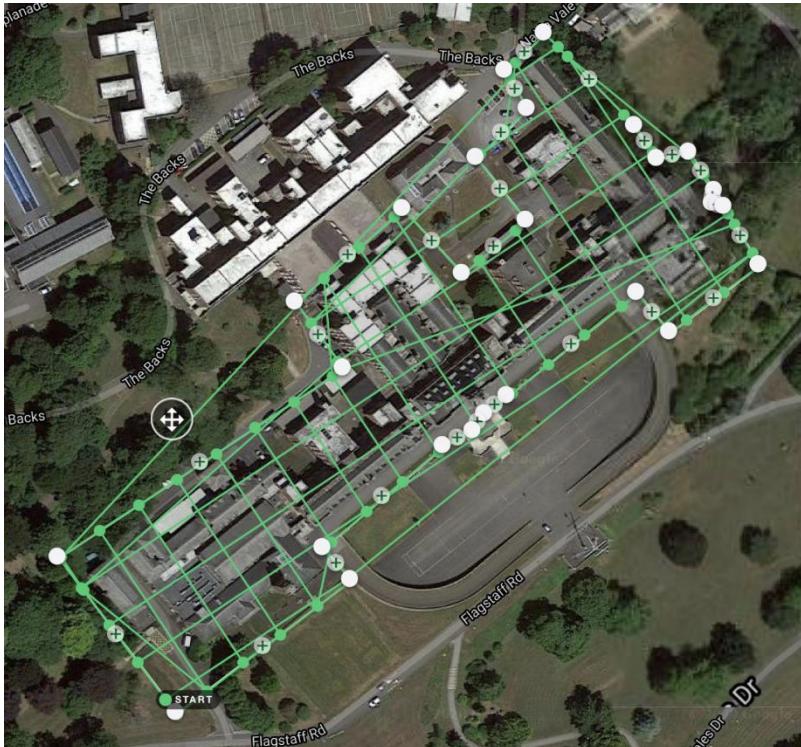
- $\lambda' = -f \frac{\Lambda z'}{z^2}$

- D'où le temps avant collision:

- $\tau = -\frac{z}{z'} = \frac{\lambda}{\lambda'}$



- Nous avons pour l'instant uniquement utilisé des paires d'image pour obtenir une information de profondeur.
- Dans le cas général, il est possible d'utiliser $N > 2$ images/caméras



- Le Bundle (block) Adjustment ou ajustement de faisceaux en bloc, est une méthode de resolution au sens des moindres carrées pour estimer la position des caméras, calculer les coordonnées 3D des points et aligner les images.
- Plusieurs images sont corrigées "en bloc".
- Principe:
 - Démarrer avec une approximation initiale
 - Projeter les points 3D sur les plans images des caméras
 - Comparaison avec la mesure
 - Ajustements pour minimiser l'erreur

- Le BA est une approche non-linéaire de résolution par moindres carrés:

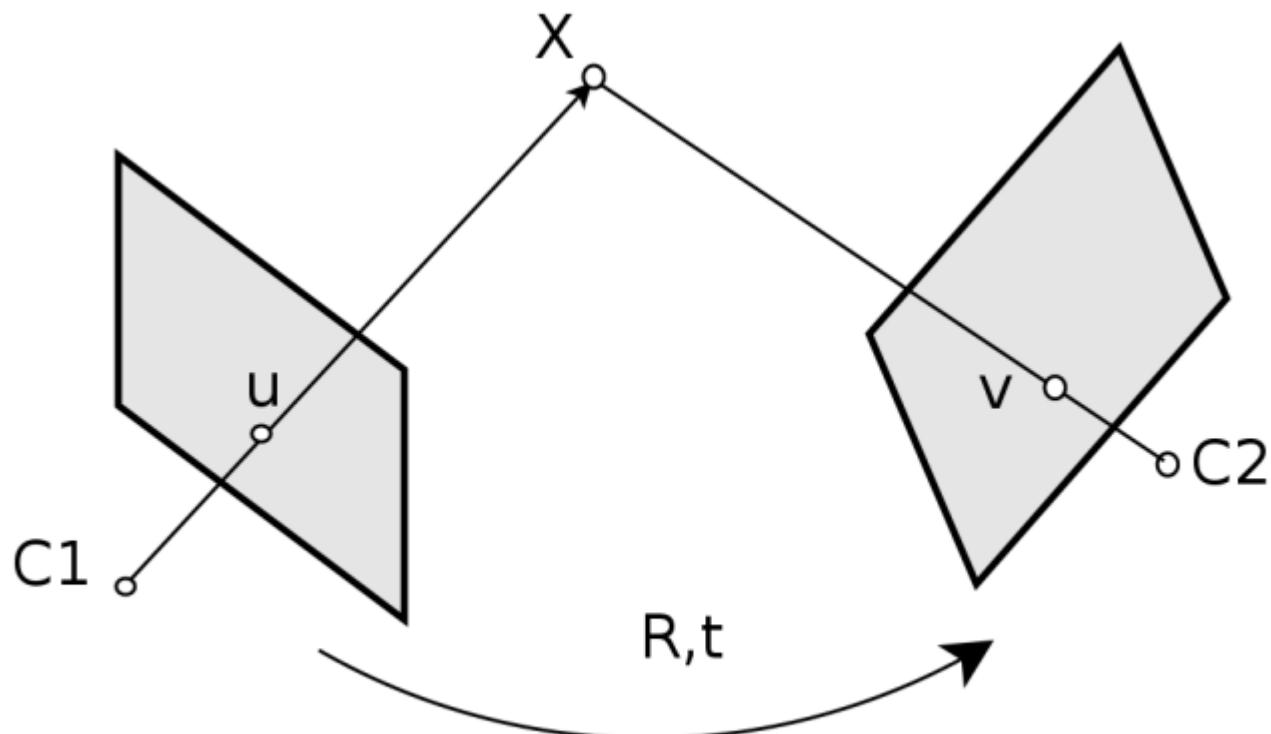
- $\bar{x}_{ij} + \hat{e}_{x_{ij}} = \lambda_{ij} P_{ij} \bar{X}_i$
 - Avec $\hat{e}_{x_{ij}}$ l'erreur de mesure du point \bar{X}_i
 - i l'indice du point, j l'indice de la caméra

- Élimination du facteur d'échelle:

- $\bar{x}_{ij} + \hat{e}_{x_{ij}} = \frac{P_{1:2}{}_{ij} \bar{X}_i}{P_3{}_{ij} \bar{X}_i}$
 - Résolution par SVD

- Estimation du mouvement de la caméra par rapport au monde
- Nécessaire à de nombreuses applications
 - Pas de GPS
 - IMU et/ou odométrie des roues insuffisants
- Odométrie:
 - Estimation du mouvement basé sur le modèle cinématique et le mouvement des roues
 - Extension à la vision

- Triangulation:
 - Permet de connaître la position 3D d'un point



- **Principe:**

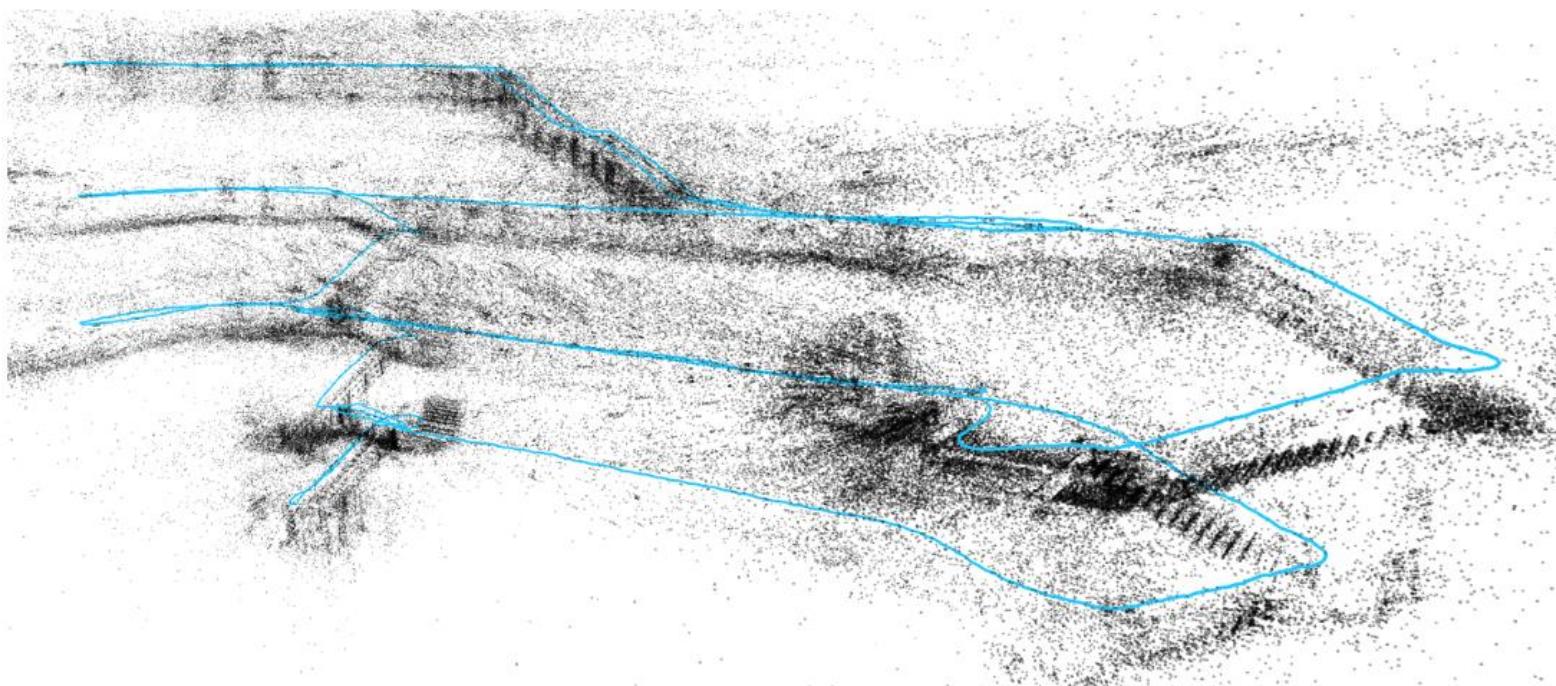
- Trouver des correspondances de points entre deux images successives: utilisation de descripteurs
- Si le monde est statique et les points bien appariés alors on peut estimer la transformation (R,t) à partir des paramètres des caméras (intrinsèques et extrinsèque)
- Problème de minimisation de l'erreur de reprojection
=> Nécessité d'une bonne calibration
- Peu robuste aux rotations pures

- Pseudo code:

- Capturer l'image I_k
- Calculer les correspondances entre I_{k-1} et I_k
- Calcul de la matrice essentielle E telle que $p^T Ep' = 0$
- Décomposition de E en R_k et t_k par SVD
- Calcul du modèle 3D (coordonnées des points de correspondance)
- Redimensionnement de t_k pour prendre en compte l'échelle: Attention! $p^T Ep' = 0 \Leftrightarrow \lambda p^T Ep' = 0$
- $k = k+1$

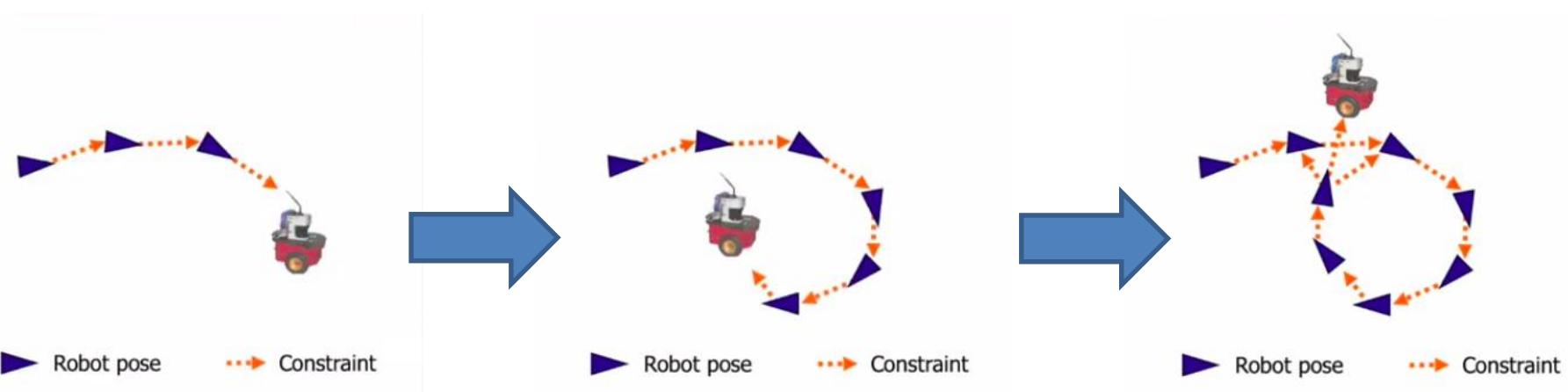
- **Simultaneous Localization And Mapping:**
 - Si une carte est fournie, possibilité de se localiser dans cette carte uniquement
 - Si une position est fournie, possibilité de créer une cartographie de l'environnement
 - Le SLAM est l'estimation conjointe d'une carte de l'environnement et de la position de la caméra dans cette carte
 - Nécessaire dès qu'un robot doit explorer un environnement totalement ou partiellement inconnu

- **Simultaneous Localization And Mapping:**
 - Amélioration de l'odométrie visuelle
 - Sauvegarde des coordonnées des points 3D extraits et de leurs caractéristiques locales
 - Création d'une carte de features 3D



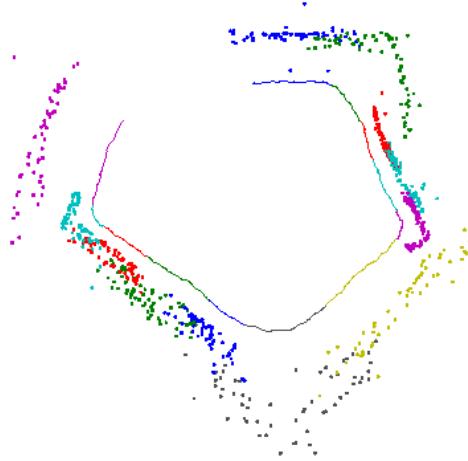
- 3 catégories principales de méthodes pour l'estimation de l'état:
 - Extended Kalman Filter
 - Particle Filter
 - Least squares => Graph-based SLAM
- Graph-based SLAM:
 - Utilisation d'un graphe pour représenter les variables et les relations entre ces variables
 - Pose Graph: contient uniquement les positions
 - Factor Graph: contient des facteurs reliant les différentes variables
 - Problème des moindres carrés résolu avec methods d'optimisation

- Pose Graph:
 - Chaque nœud représente une pose
 - Les liaisons entre ces nœuds contiennent leur relation spatiale
 - L'optimisation essaye de trouver la position optimale d'un nœud qui minimise l'erreur introduite dans les liaison

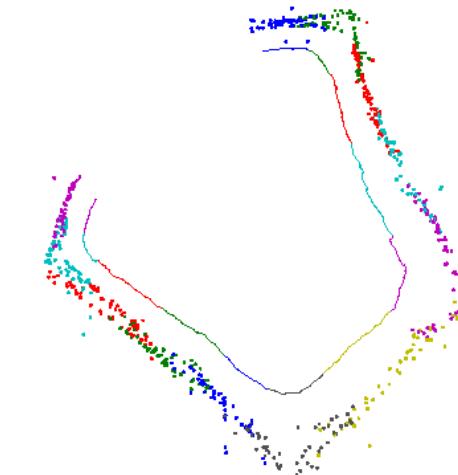


- Quels sont les avantages?

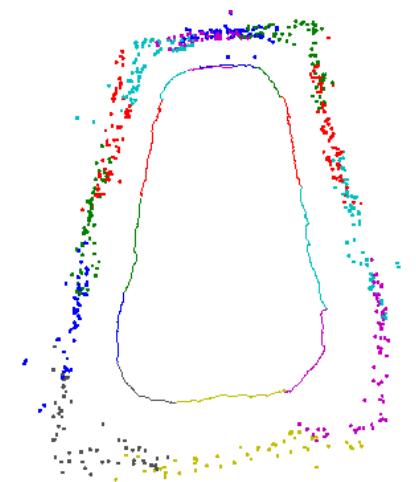
- Meilleure estimation des coordonnées 3D des points/features
- Fermeture de boucles (Loop-closure)
- Plus robuste face aux rotations pures



a) Local maps obtained with pure monocular LAM



(b) Local maps auto-scaled



(c) After loop closing

Merci pour votre attention

Questions?