

Descripteurs et mises en correspondance

Cours ENSTA Paris – ROB317

David FILLIAT

david.filliat@ensta-paris.fr

2019-2020



Objectifs du cours

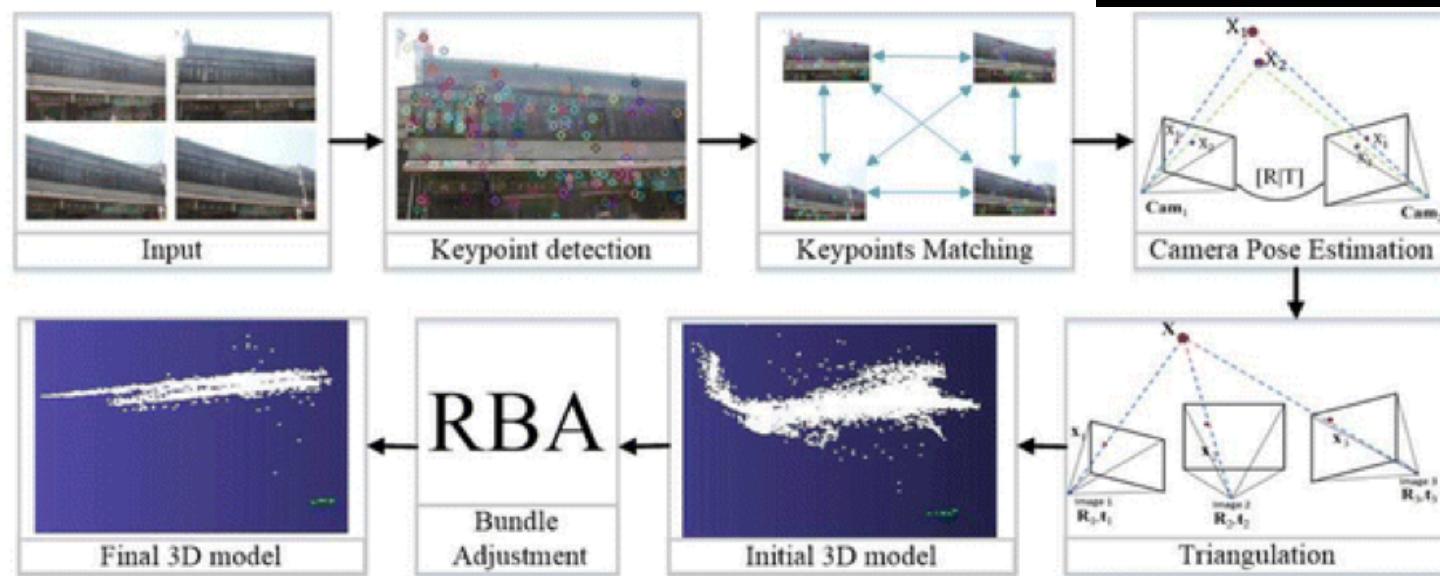
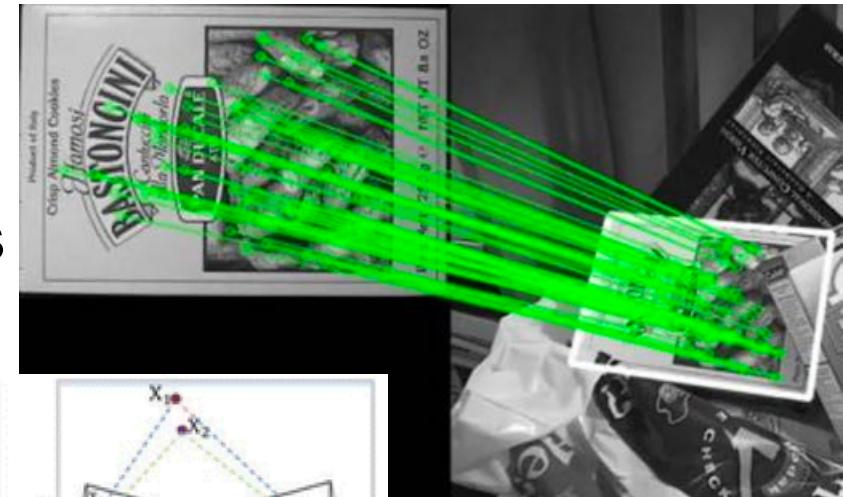
- Comment représenter l'apparence de points d'intérêt ?
- Comment garantir des invariances ?



- Comment trouver efficacement des correspondances ?

Intérêt applicatif

- Reconnaissance d'objets texturés
- Reconstruction 3D, SLAM
- Odométrie visuelle
- Reconstruction de panoramas
- ...



Descripteurs de points d'intérêt

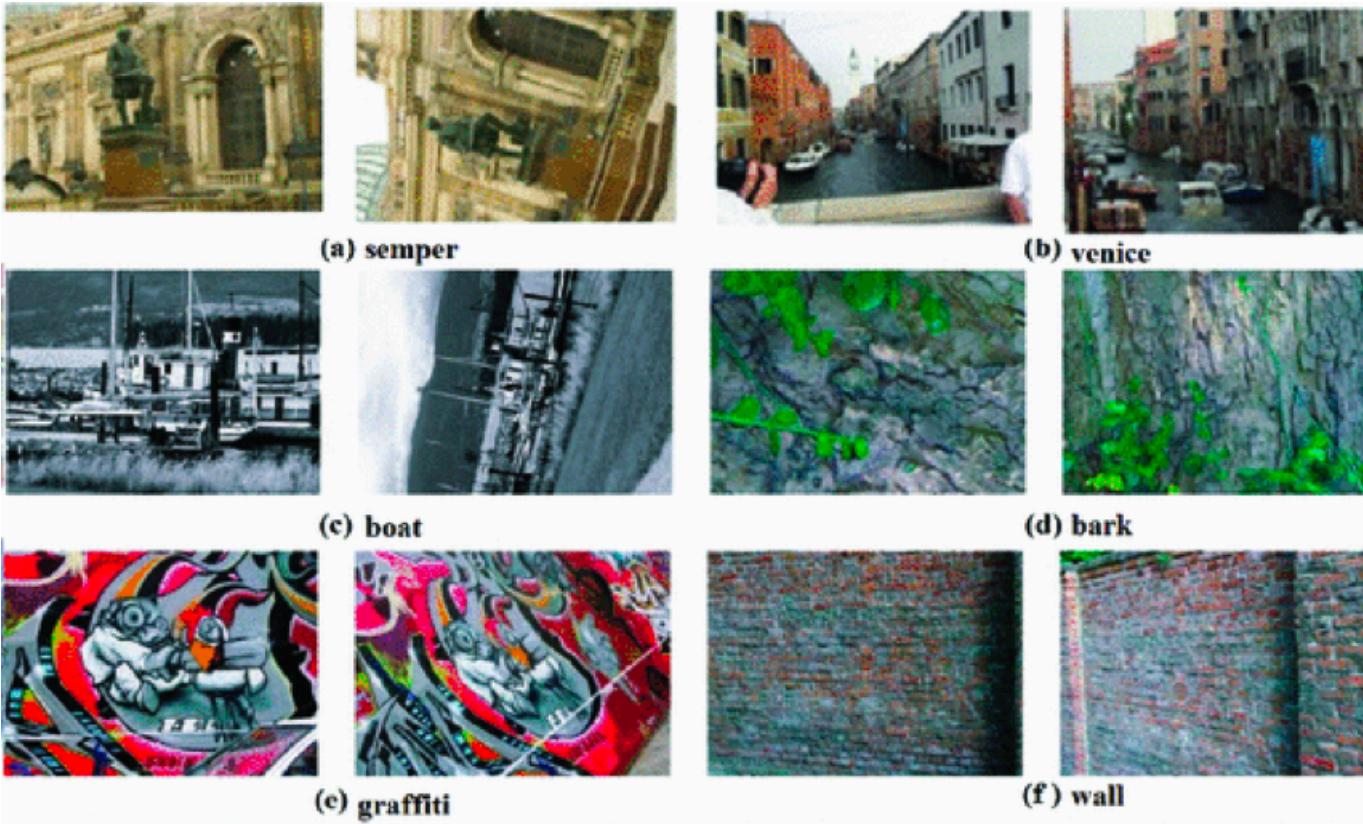
- Imagettes
- Invariant différentiels
- Descripteurs SIFT
- Descripteurs rapides : SURF, U-SURF
- Descripteurs binaires : BRIEF, ORB

Mise en correspondances de points d'intérêt

- Force brute, reverse matching, distance ratio
- KD trees, approximate kd-trees
- Hierarchical k-means, bag of words

Invariances ?

- Luminosité / contraste
- Translation / rotation / échelle dans le plan
- Transformations 3D, rotations hors du plan (point de vue)



Imagettes (Image patch)

- Représentation la plus simple
- Choisir une taille suffisamment informative / calcul rapide



- Hypothèse de décalage en translation
 - Comment comparer $I_1(x_1, y_1)$ et $I_2(x_2, y_2)$?
 - Robustesse changement luminosité / contraste ?

« Sum of Squared Differences »

$$SSD(I_1, x_1, y_1, I_2, x_2, y_2) = \sum_{i=-n}^n \sum_{j=-p}^p (I_1(x_1 + i, y_1 + j) - I_2(x_2 + i, y_2 + j))^2$$

Cross corrélation normalisée

- Invariance / luminosité et contraste

$$NCC(I_1, x_1, y_1, I_2, x_2, y_2) = \frac{\sum_{i,j} (I_1(x_1 + i, y_1 + j) - \overline{I_1(x_1, y_1)})(I_2(x_2 + i, y_2 + j) - \overline{I_2(x_2, y_2)})}{\sqrt{\sum_{i,j} (I_1(x_1 + i, y_1 + j) - \overline{I_1(x_1, y_1)})^2 \sum_{i,j} (I_2(x_2 + i, y_2 + j) - \overline{I_2(x_2, y_2)})^2}}$$

Avantages / Inconvénients

- Proche des données, rapide à calculer,
- Faible invariance aux déformations
- Pas d'invariance en rotation, invariance échelle par norm.
- Adapté aux faibles changements de pt de vue

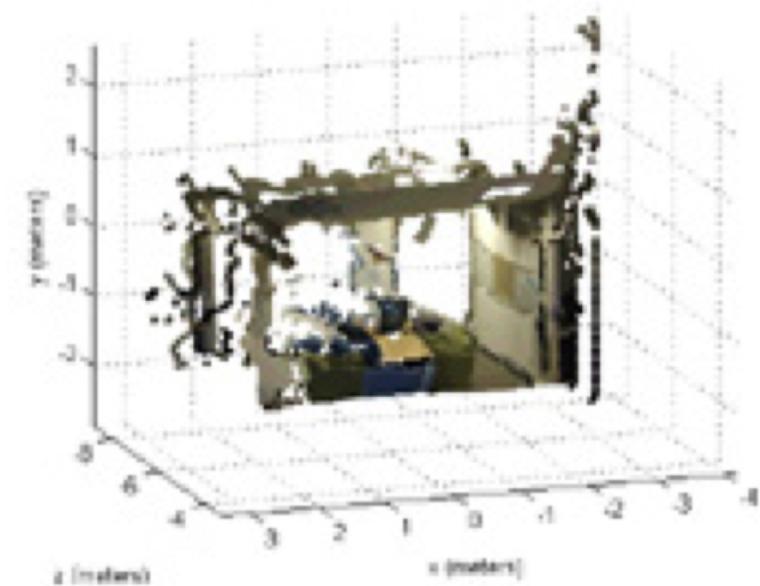
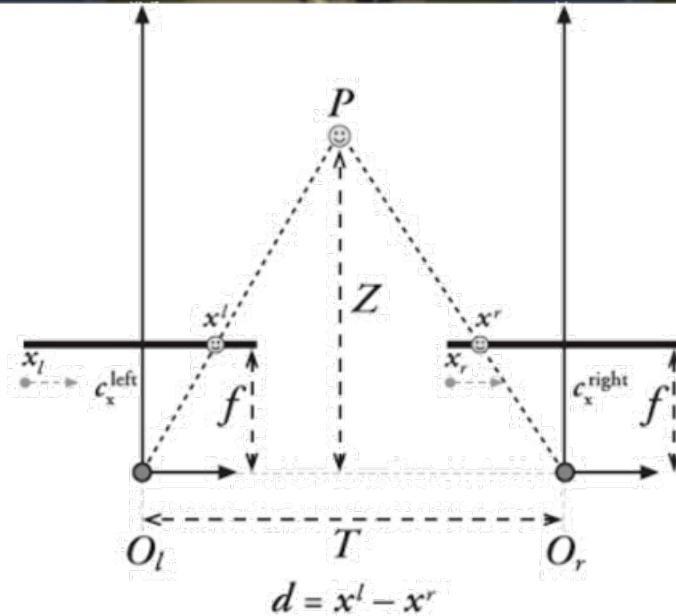
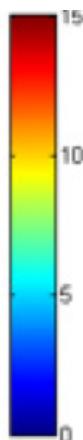
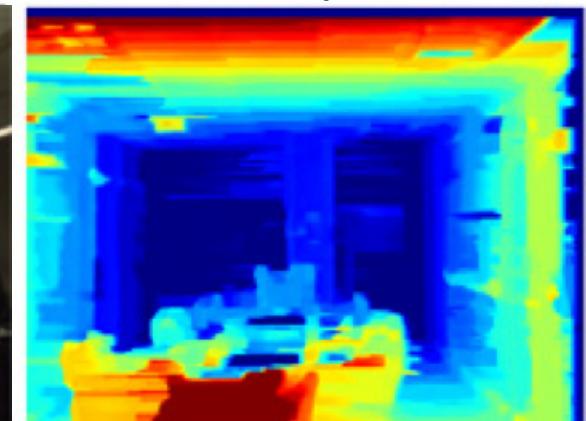
Ex application : stéréovision

Gauche



Droite

Disparité



Invariant différentiels

- Exploite les dérivées spatiales multi-échelles
- Construit des indices invariant à la rotation

« Jet local »

$$L_{ij}^{\sigma} = I * G_{ij}^{\sigma} \quad \text{avec : } G_{ij}^{\sigma} = \frac{\partial^{i+j}}{\partial x^i \partial y^j} G^{\sigma} \quad \text{et : } G^{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

- Par exemple, dérivées jusqu'à ordre 3:

$$\{L_{ij}^{\sigma}; 0 \leq i + j \leq 3\} = \{L, L_x, L_y, L_{xx}, L_{xy}, L_{yy}, L_{xxx}, L_{xxy}, L_{xyy}, L_{yyy}\}$$

Combinaisons invariantes

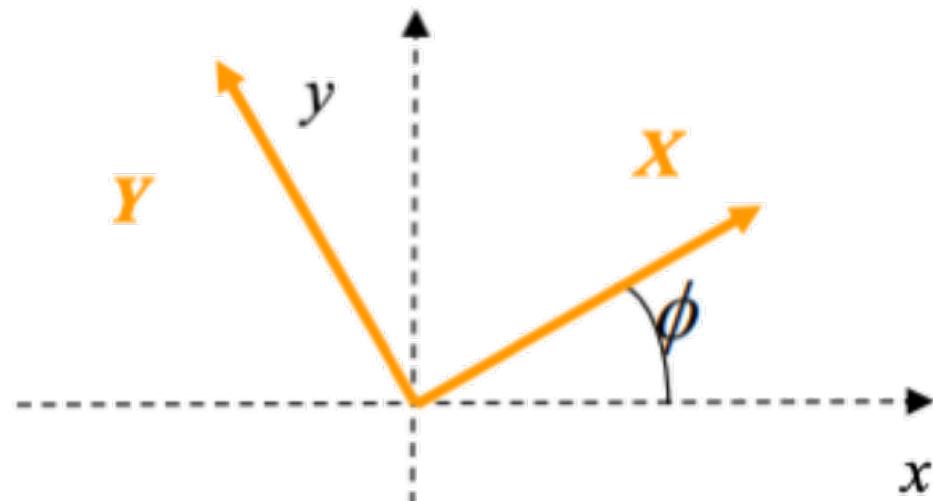
– Par exemple, le laplacien est invariant:

$$\begin{cases} x = X \cos \phi + Y \sin \phi \\ y = X \sin \phi - Y \cos \phi \end{cases}$$

$$\begin{cases} X = x \cos \phi + y \sin \phi \\ Y = -x \sin \phi + y \cos \phi \end{cases}$$

$$\begin{cases} I_X = I_x \cos \phi + I_y \sin \phi \\ I_Y = I_x \sin \phi - I_y \cos \phi \end{cases} \quad \text{et :} \quad \begin{cases} I_{XX} = I_{xx} \cos^2 \phi + 2I_{xy} \cos \phi \sin \phi + I_{yy} \sin^2 \phi \\ I_{YY} = I_{xx} \sin^2 \phi - 2I_{xy} \cos \phi \sin \phi + I_{yy} \cos^2 \phi \end{cases}$$

et donc : $I_{XX} + I_{YY} = I_{xx} + I_{yy}$



Invariants différentiels de Hilbert

$$\Psi = \left(\begin{array}{c} L \\ L_i L_i \\ L_i L_{ij} L_j \\ L_{ii} \\ L_{ij} L_{ij} \\ \varepsilon_{ij} (L_{jkl} L_i L_k L_l - L_{jkk} L_i L_l L_l) \\ L_{ijk} L_j L_k L_k - L_{ijk} L_i L_j L_k \\ - \varepsilon_{ij} L_{jkl} L_i L_k L_l \\ L_{ijk} L_i L_j L_k \end{array} \right)$$

Avec

$$\begin{aligned}\varepsilon_{xx} &= \varepsilon_{yy} = 0 \\ \varepsilon_{xy} &= -\varepsilon_{yx} = 1\end{aligned}$$

Notations d'Einstein
(somme sur les indices)
par ex :

$$\begin{aligned}\Psi_2 &= L_i L_{ij} L_j = L_x L_{xx} L_x + 2L_x L_{xy} L_y + L_y L_{yy} L_y \\ \Psi_7 &= -\varepsilon_{ij} L_{jkl} L_i L_k L_l = L_{xxy} (-L_x L_x L_x + 2L_x L_y L_y) \\ &\quad + L_{xyy} (-2L_x L_x L_y + L_y L_y L_y) - L_{yyy} L_x L_y L_y + L_{xxx} L_x L_x L_y\end{aligned}$$

Comparaison des invariants

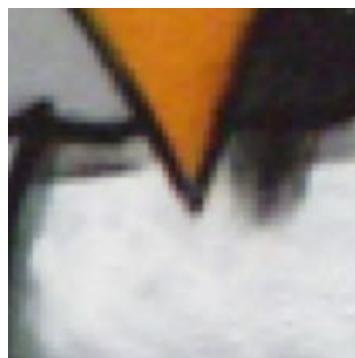
- Calcul des invariants à partir de l'échelle du point d'intérêt
- Vecteur des invariants à plusieurs échelles relatives
- Comparaison avec norme L2

Avantages / Inconvénients

- Robustesse luminosité/contraste
- Invariance rotation par construction
- Invariance échelle grâce à échelle de référence du point
- Peu utilisés en pratique ?

Descripteurs SIFT (Scale Invariant Feature Transform)

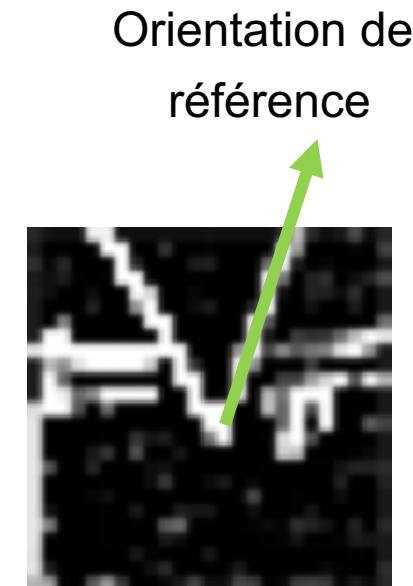
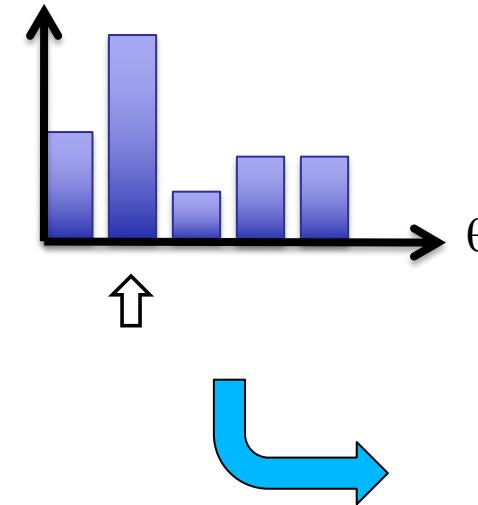
- Associé au détecteur SIFT
- Histogramme des orientations du gradient
- Utilise les gradients à l'échelle du point détecté
- Orientation de référence : max des directions de gradient



Image

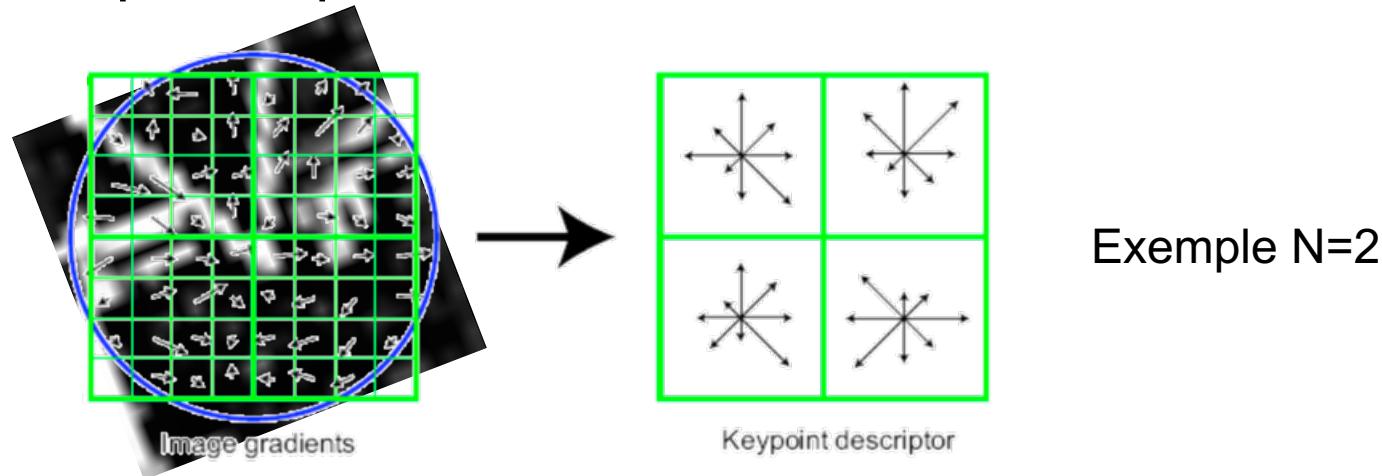


Gradient



Histogramme d'orientation du gradient

- Directions relatives à l'orientation de référence
- Découpage en $N^2 \times 4 \times 4$ carrés
- Histogramme en 8 bins des orientations de chaque carré
- Pondéré par amplitude et inverse distance au centre



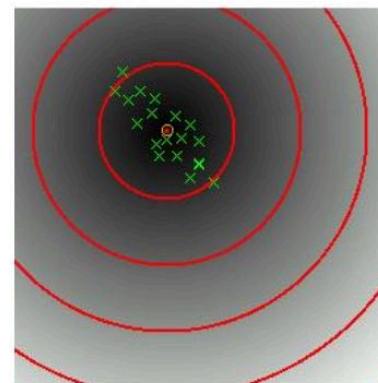
- Descripteur : hist. concaténés/normalisés : $N^2 \times 8$ réels
- Valeur classique N=4, descripteur de taille 128
- Normalisé tels que $\sum d_i^2 = 1 \quad d_i < 0.2$

Comparaison SIFT

- Comparaison standard avec norme L2

L2

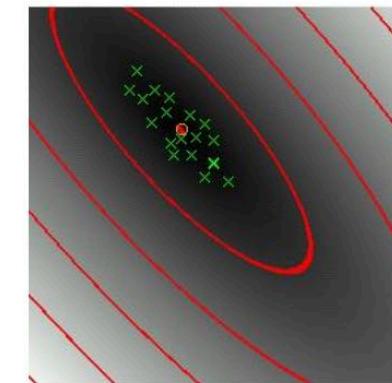
$$(x - x')^T (x - x')$$



Mahalanobis

$$(x - x')^T C^{-1} (x - x')$$

$$C = (\text{cov}(x_i, x_j))_{i,j}$$



- Grande base de donnée: possibilité d'exploiter les corrélations entre dimensions : distance de Mahalanobis

Avantages / Inconvénients

- Robustesse luminosité/contraste/rotation/échelle
- Invariance raisonnable aux déformations 3D
- Parmis les meilleures performances
- Relativement lourd à calculer -> applications 'offline'

Descripteurs SURF (Speeded-Up Robust Features)

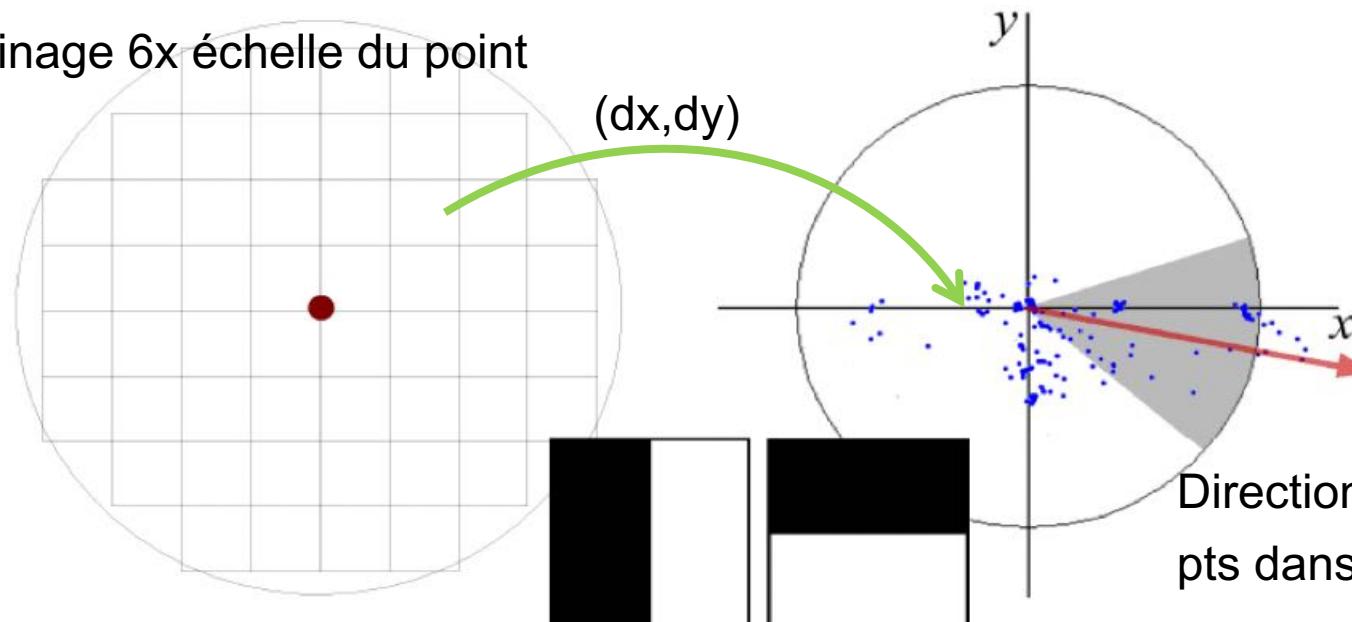
- Associé au détecteur SURF (et KAZE)
- Exploite les images intégrales :

Ondelettes de Haar pour calculer le gradient



Orientation de référence

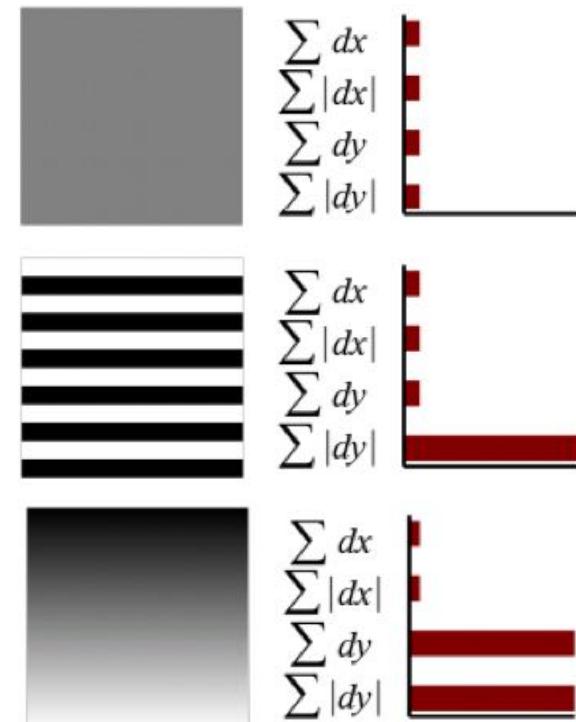
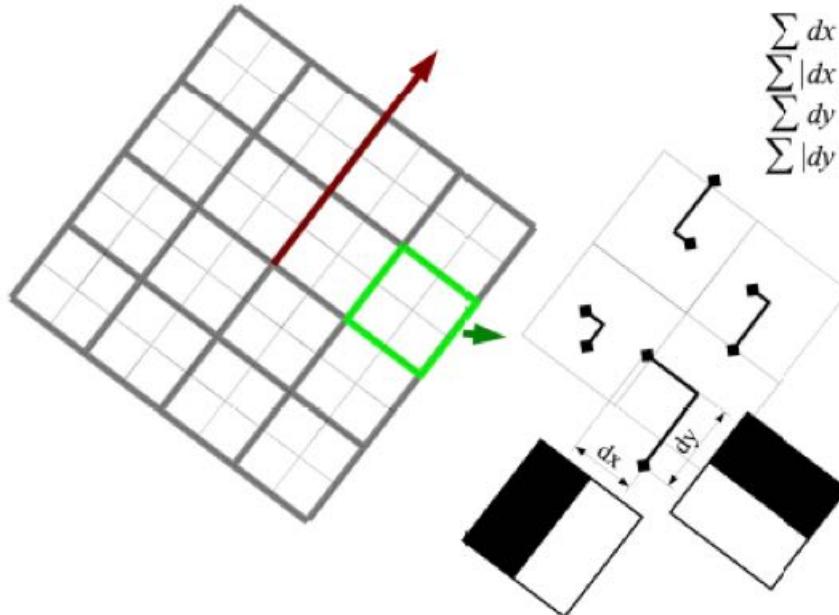
Voisinage 6x échelle du point



Direction avec max de pts dans un cône de 60°

Descripteurs

- Grille 4x4 cases, 5x5 points/case
- Pour chaque case, somme des dx , $|dx|$, dy , $|dy|$



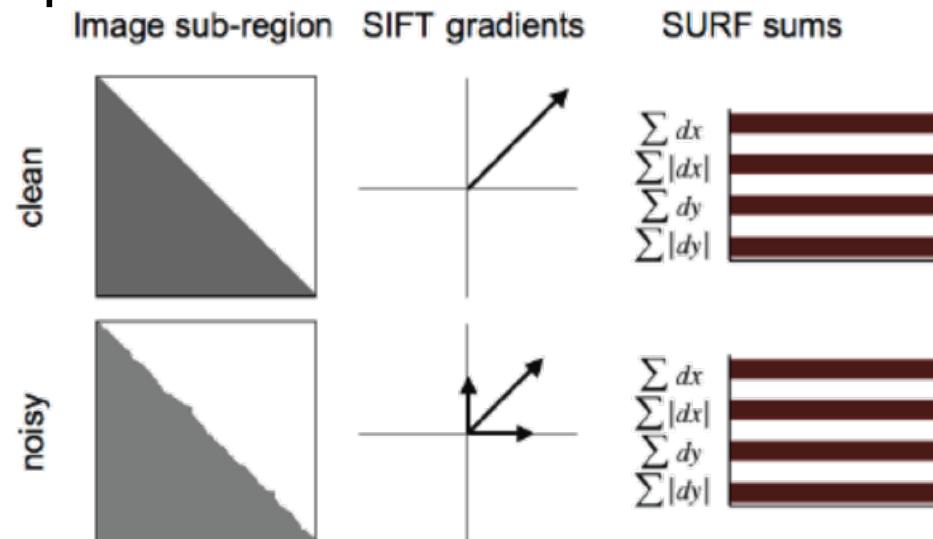
- Descripteur dimension $64=4\times 4\times 4$
- Version U-SURF : sans orientation de référence

Comparaison SURF

- Comparaison avec norme L2 / Mahalanobis

Avantages / Inconvénients

- Robustesse luminosité/contraste/rotation/échelle
- Invariance raisonnable aux déformations 3D
- Performances proches de SIFT / calcul plus léger
- Meilleur que SIFT sur certains cas

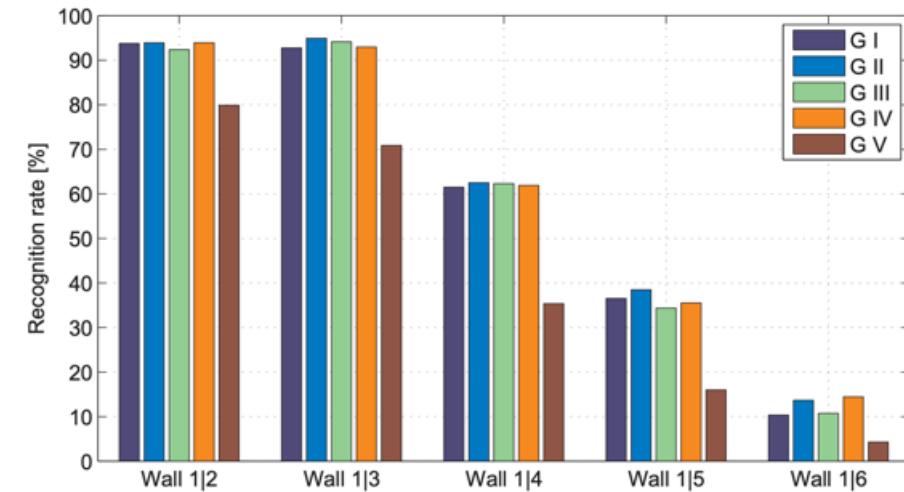
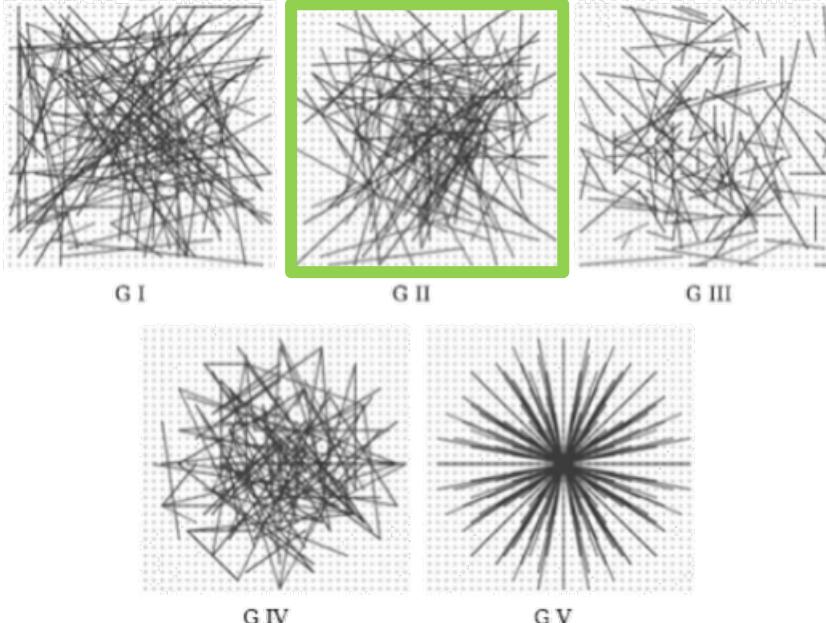


Descripteurs binaires

- Descripteurs adaptés à l'embarqué, compacts et rapides
- Ex : BRIEF : 256 bits vs SIFT : 128 réels

BRIEF: binary robust independent elementary features

- Résultats de comparaisons de niveau de gris
- Paires de pixels comparés choisis aléatoirement

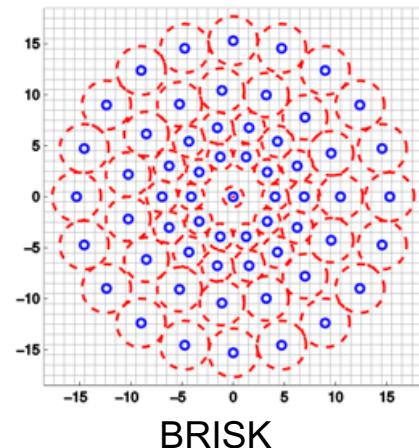
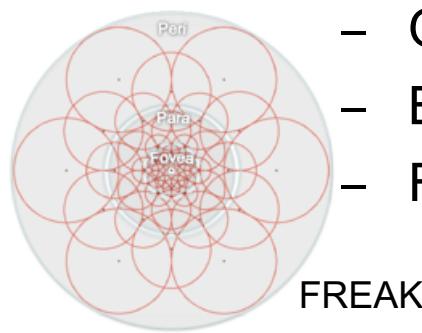


Comparaison BRIEF

- Distance de Hamming : $\text{Sum}(\text{XOR}(x, x'))$
- Rapide avec extension SIMD processeur (SSE, Neon)

Avantages/inconvénients

- Très rapide / bonnes performances
- Pas adapté à l'accélération des comparaison par arbres
- Pas invariant en rotation, mais peut être associé à un détecteur orienté
- Variantes/extensions :
 - ORB : apprendre les paires à comparer
 - BRISK : tests entre zones circulaires
 - FREAK : zones inspirées de la rétine



Performances des descripteurs

- Issu de « An Experimental Evaluation of Binary Feature Descriptors, Alshazly et al. »

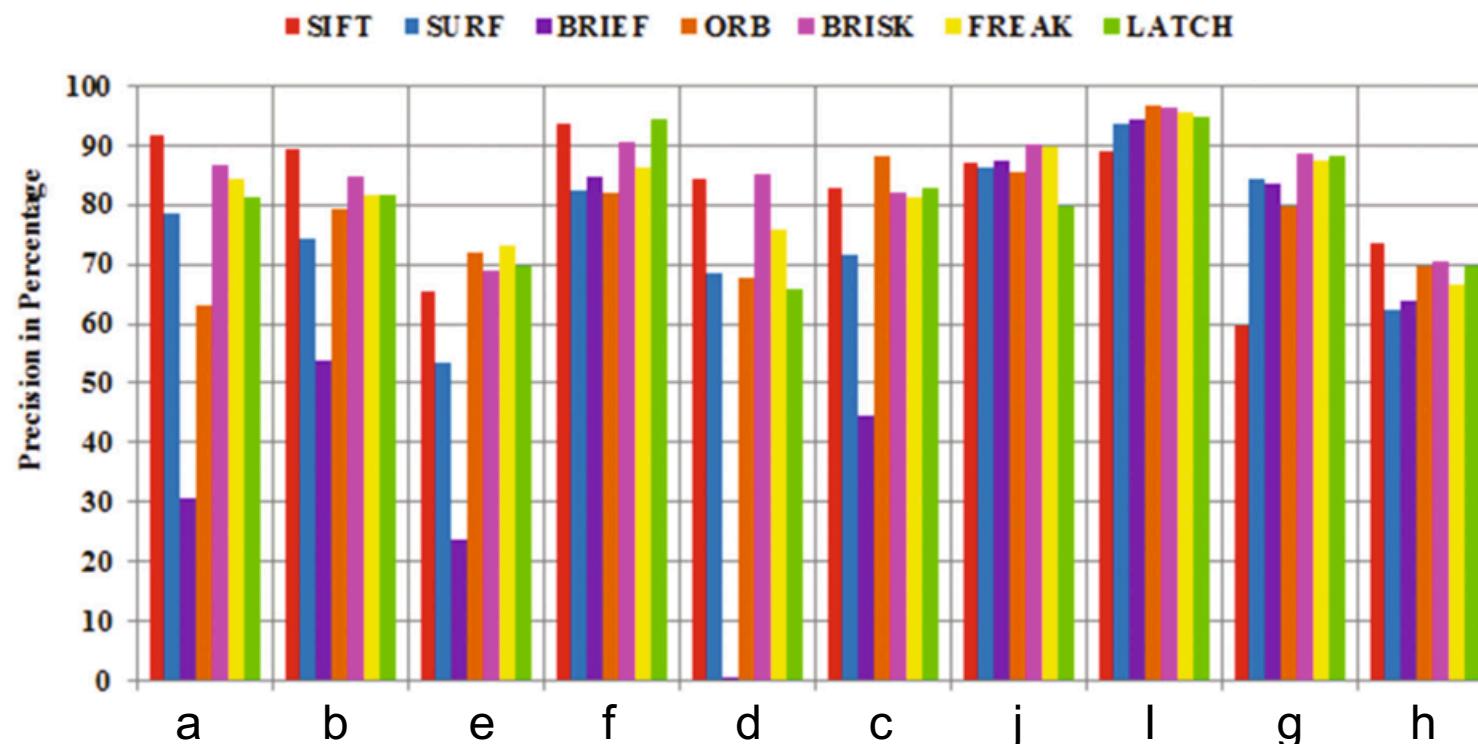


Fig. 1. Sample images from the datasets used in the evaluation: (a) pure rotation, (b) pure scaling, (c) and (d) combined scale and rotation, (e) and (f) viewpoint changes, (g) and (h) blur, (i) JPEG compression, and (j) illumination changes.

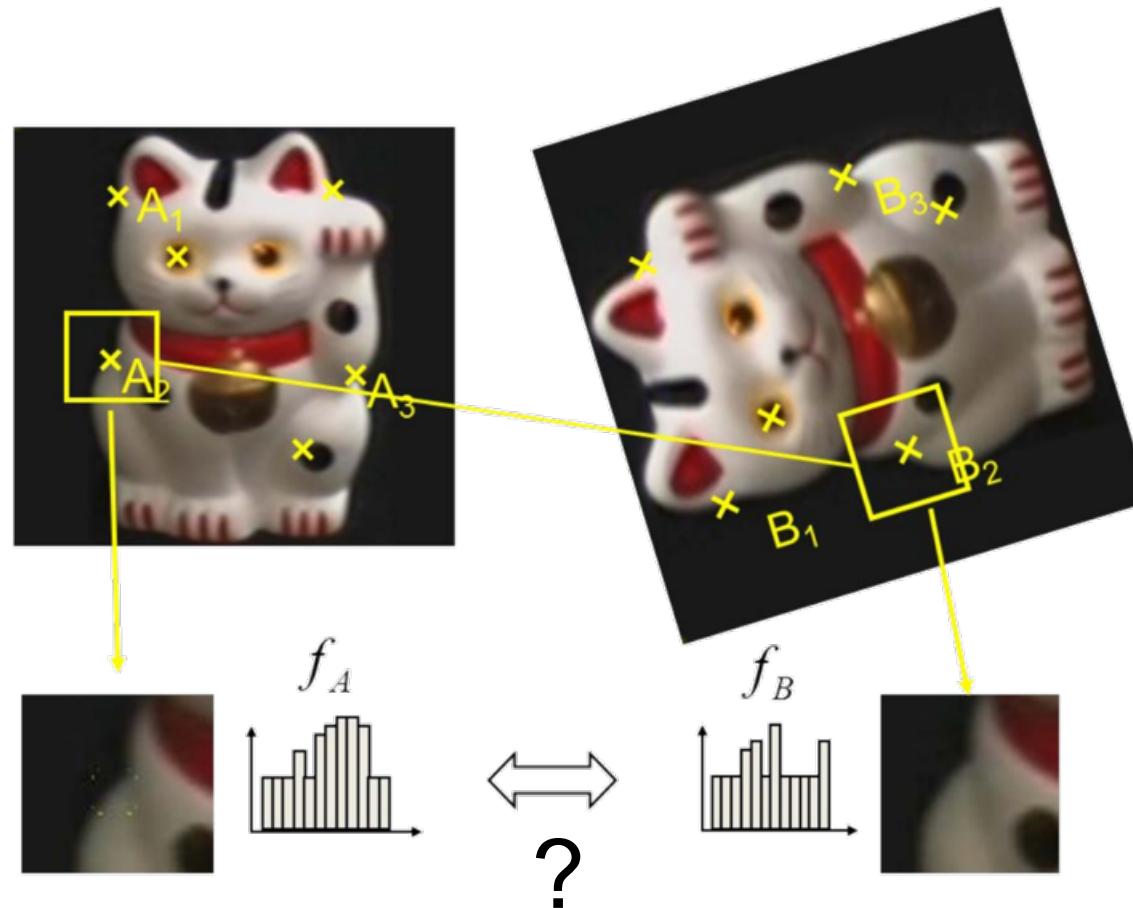
Performances des descripteurs

- Issu de « An Experimental Evaluation of Binary Feature Descriptors, Alshazly et al. »

Descriptor	BRIEF	ORB	BRISK	FREAK	LATCH	SIFT	SURF
Extraction time (ms)	0.20586	0.21100	0.09097	0.47647	0.86376	1.78218	1.05287
Memory size (bytes)	32	32	64	64	32	128(512)	64(256)

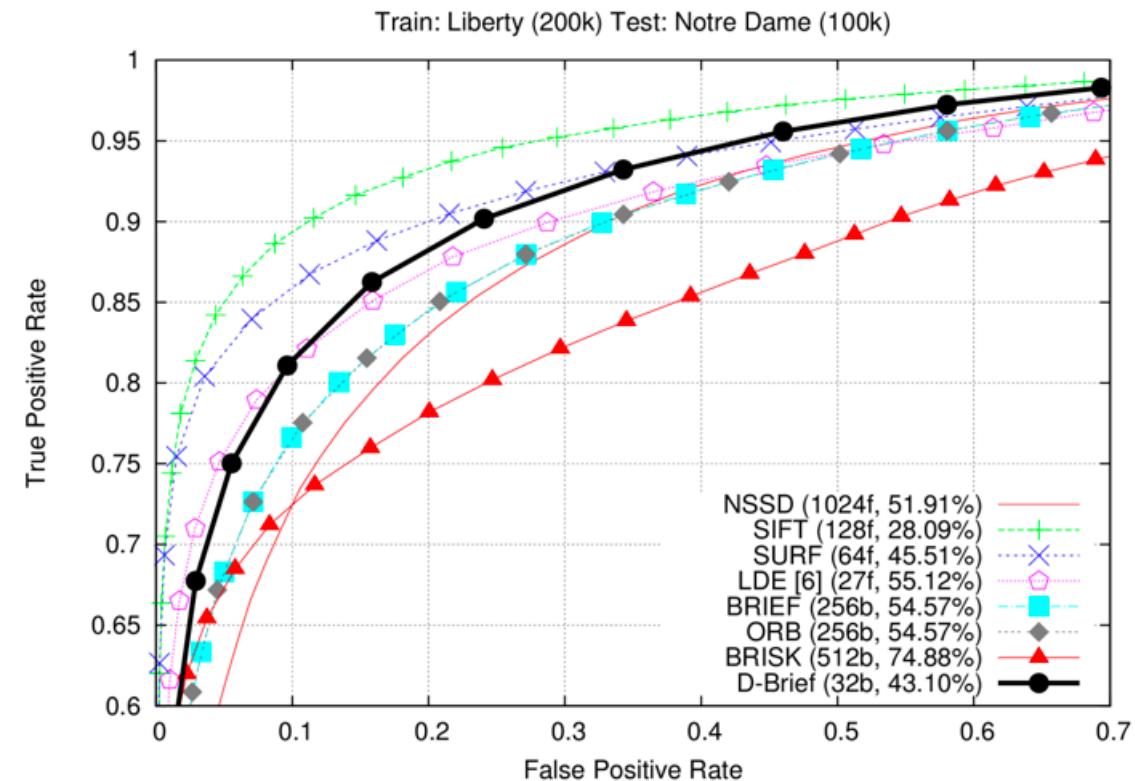
Objectif

- Trouver les points correspondants entre deux images



Force Brute

- Comparaison du descripteur à chaque descripteur de l'image cible
- Conserver le plus proche si distance < seuil
- Voir courbe ROC pour choix du seuil



<https://www.epfl.ch/labs/cvlab/research/descriptors-and-keypoints/research-detect-dbrief/>

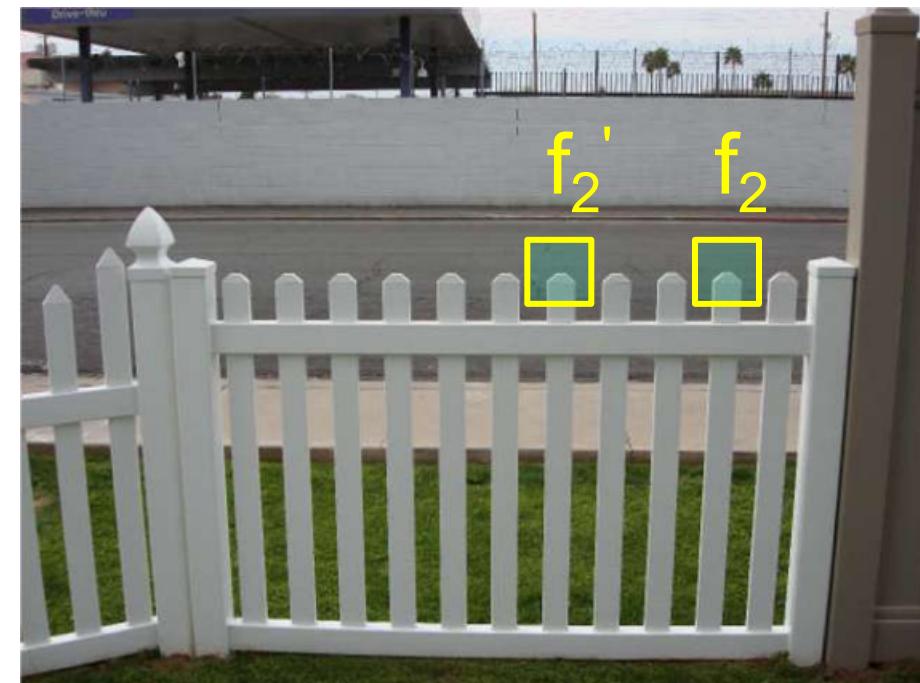
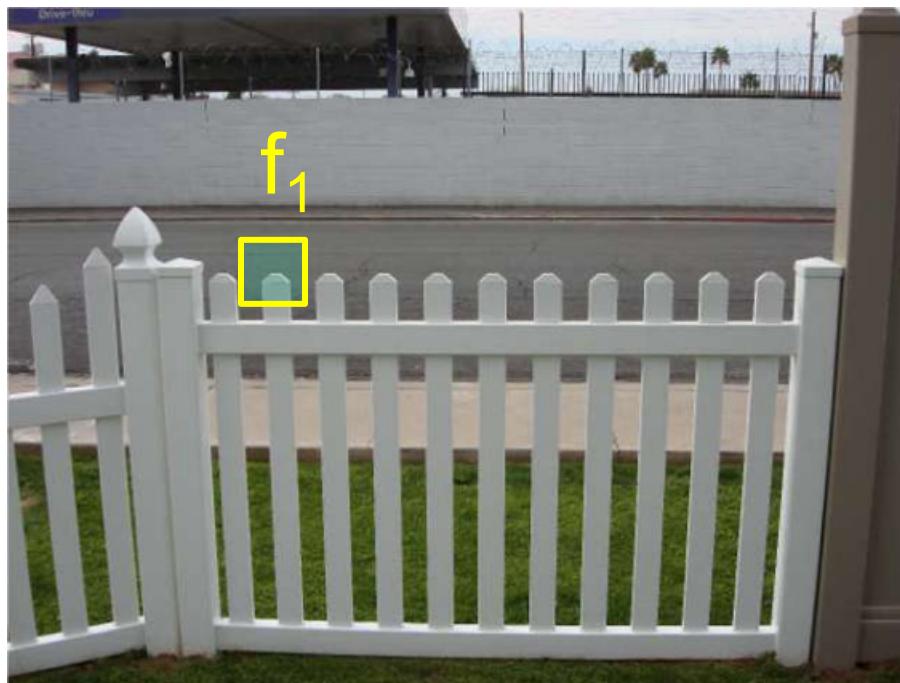
Reverse matching (cross check)

- Effectuer les correspondances dans les deux sens, conserver en cas d'accord
- Permet de réduire les erreurs sur les pts absents



Distance ratio

- Rapport des distances vers les deux descripteurs les plus proches

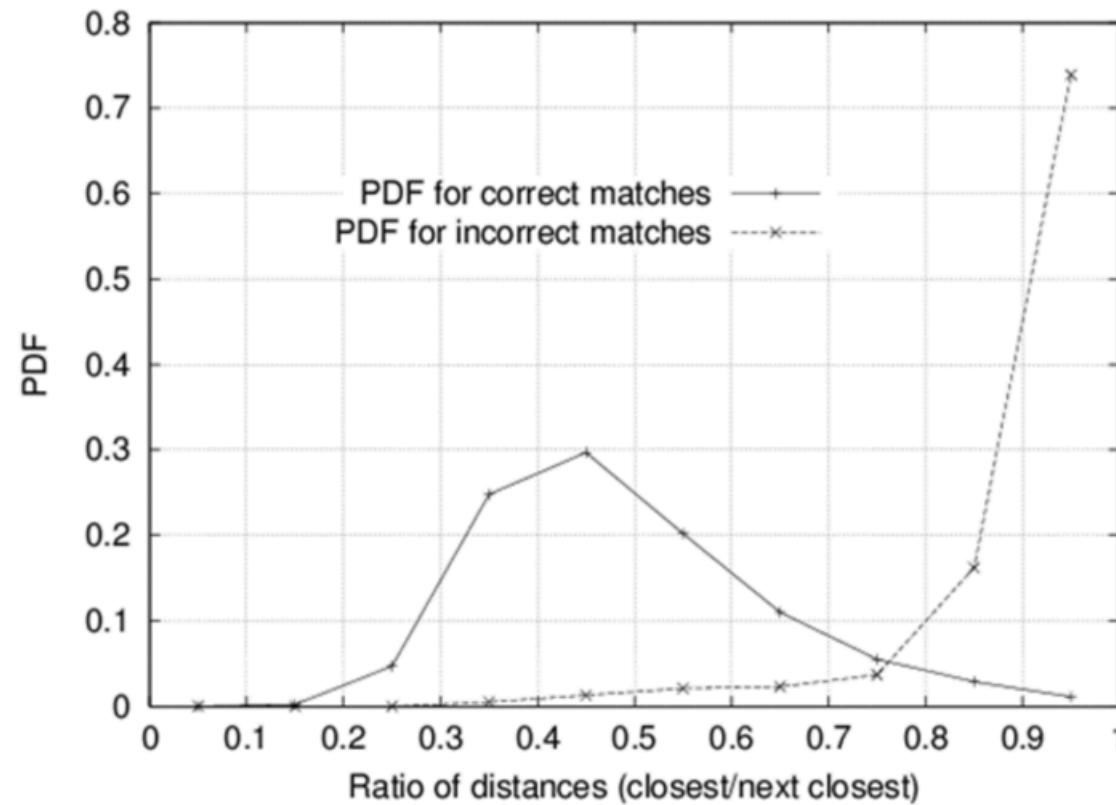


$$– r = d(f_1, f_2) / d(f_1, f_2')$$

ici : $r \sim 1 \rightarrow$ ambiguïté

Distance ratio

- Conserver matching avec $r < 0.8$



David G. Lowe. "Distinctive image features from scale-invariant keypoints." IJCV 60 (2), pp. 91-110, 2004

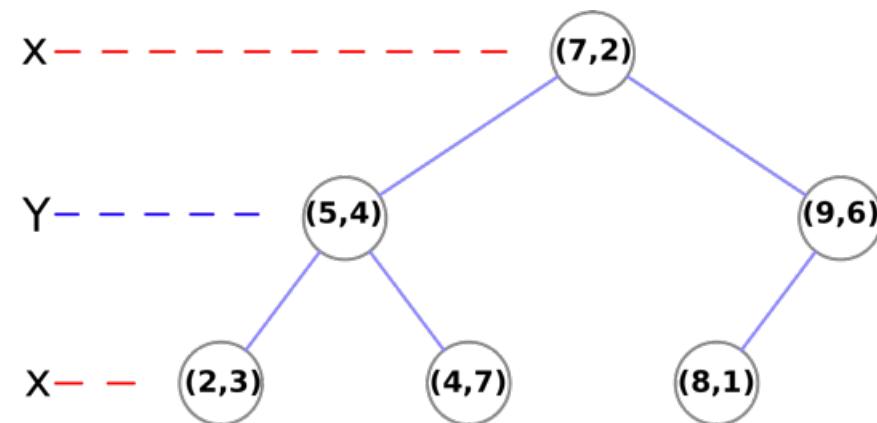
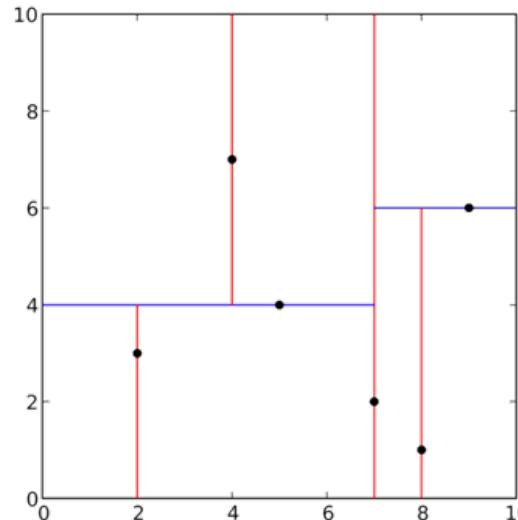
Force Brute

KD Trees

- K-dimensional trees
 - Accélérer recherche des plus proches voisins
 - Arbre binaire équilibré
 - Découpe récursivement l'espace en alternant les directions

KD Trees : création

- Couper récursivement les données en 2 parties
- Couper sur la médiane des points
- Alterner les dimension de découpe (1,2 ...N, 1, 2 ...)

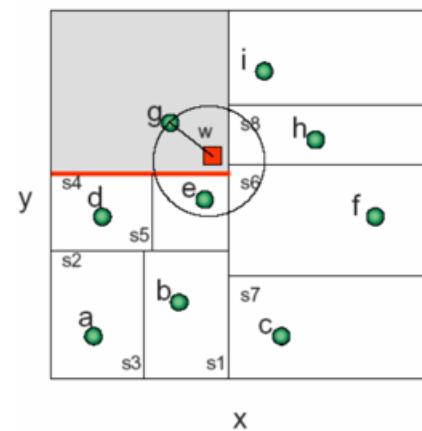


- Variantes :
 - Découper dans la direction de plus grande variance
 - Direction aléatoire parmi le plus grandes variances
 -

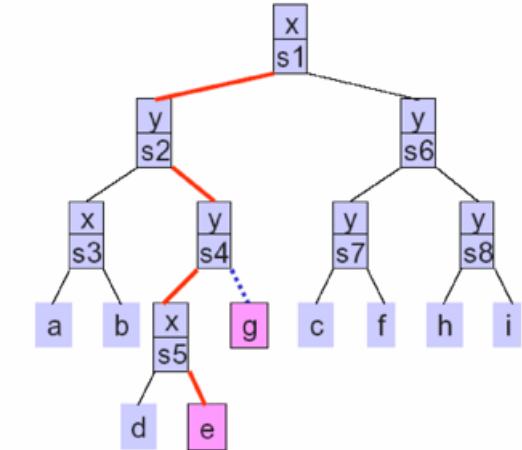
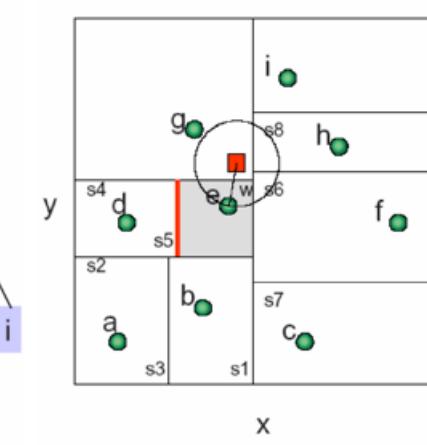
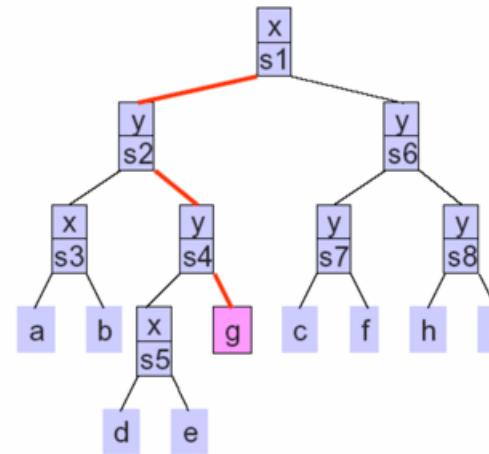
KD Trees : recherche

- Descendre dans l'arbre pour trouver le plus proche voisin dans la branche
- Remonter et redescendre les feuilles intersectant la sphère du plus proche voisin
- Performance moyenne : $\log(N)$
- **Attention** : faibles intérêt avec dist. de Hamming

■ query point



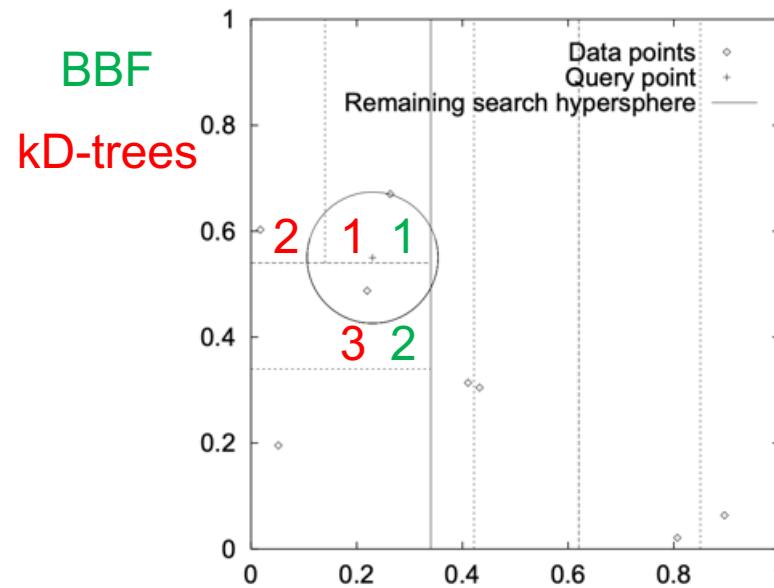
■ query point



Puis explorer s6, S8...

Approximate kd-trees (Best Bin First Search)

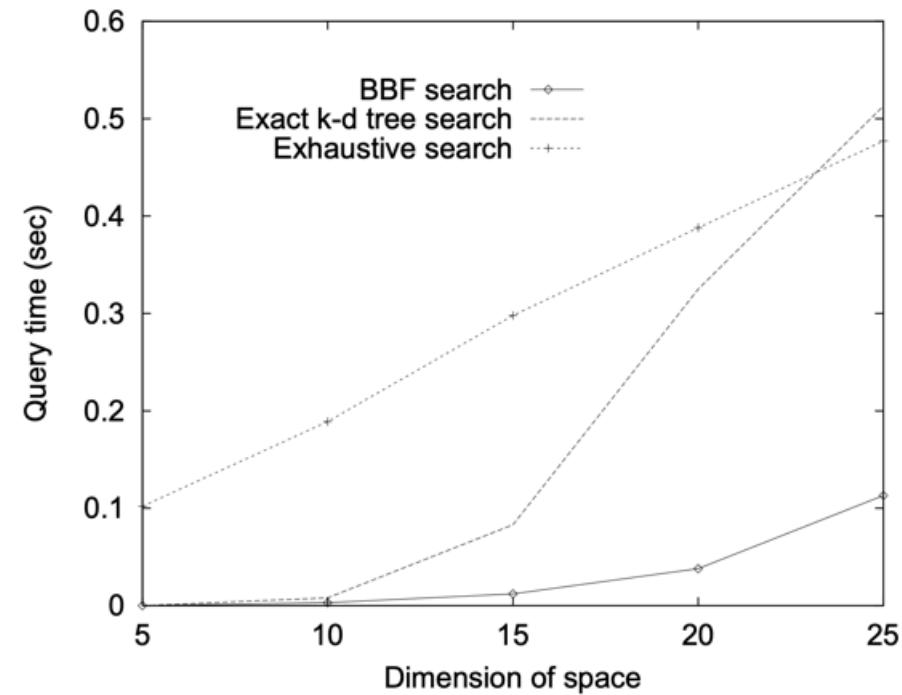
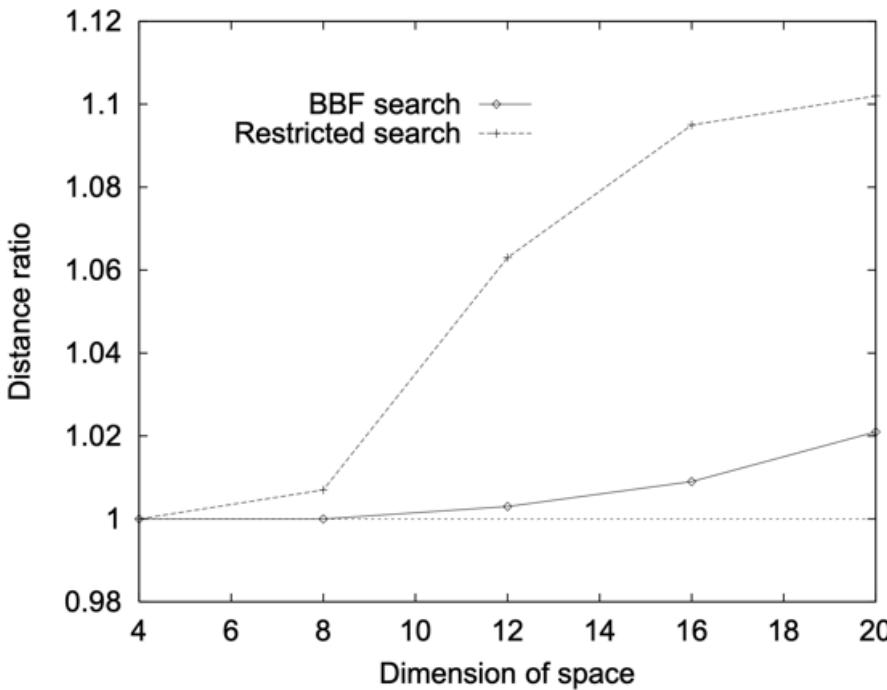
- En grande dimensions (e.g. 128 for SIFT), KD-trees sont pire que recherche brute car toutes les feuilles doivent être examinées
- BBF examine les feuilles les plus proches du point en premier (et non en parcourant l'arbre)
- BBF limite les nombre de feuilles examinées



Beis, J.; Lowe, D. G. (1997). *Shape indexing using approximate nearest-neighbour search in high-dimensional spaces*. CVPR

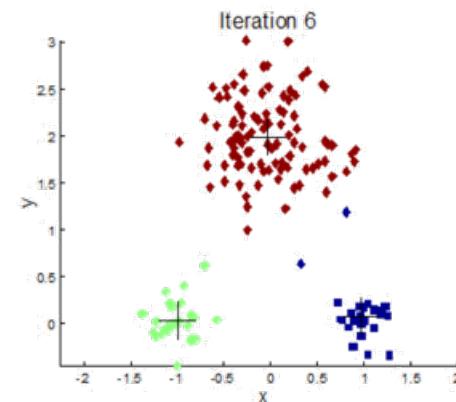
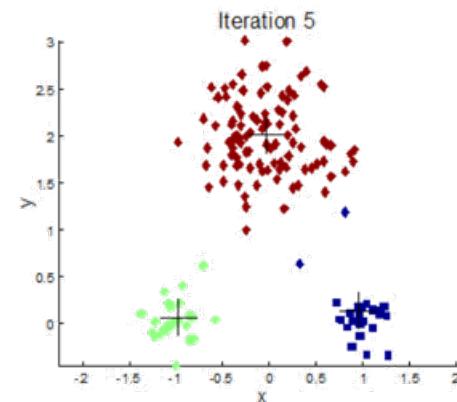
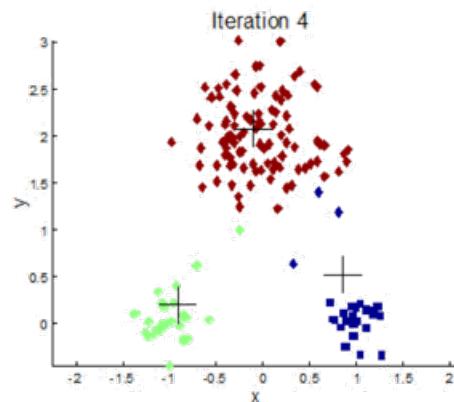
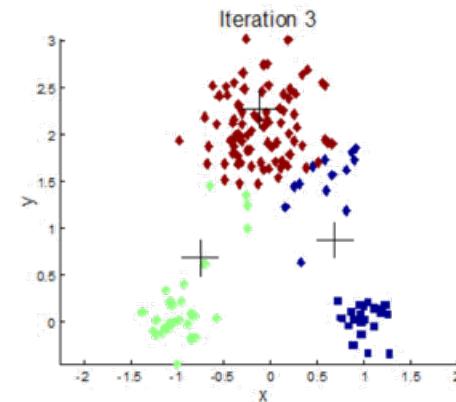
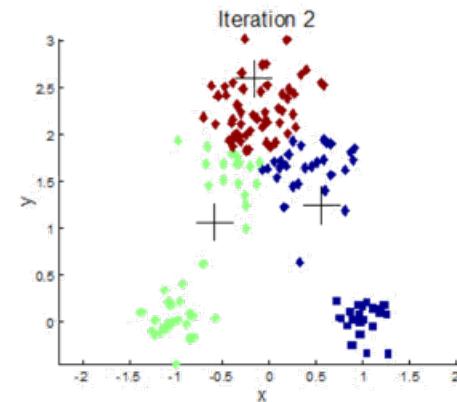
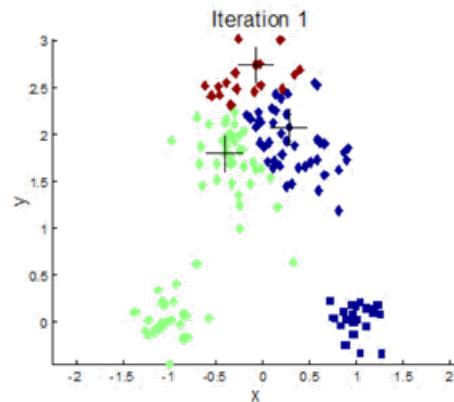
Approximate kd-trees (Best Bin First Search)

- Points trouvés restent proche du vrai plus proche voisin
- Gain de performances significatif dès que $\text{dim} > 25$



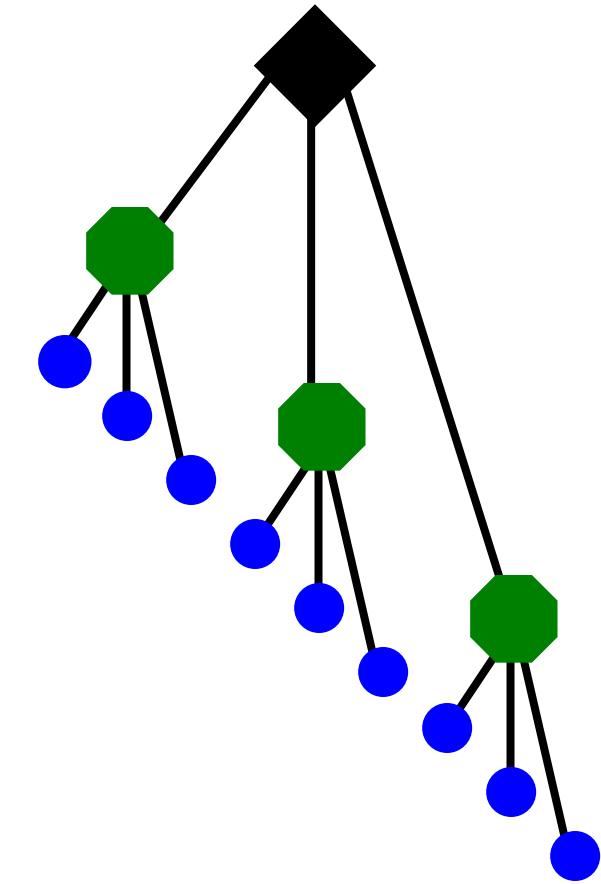
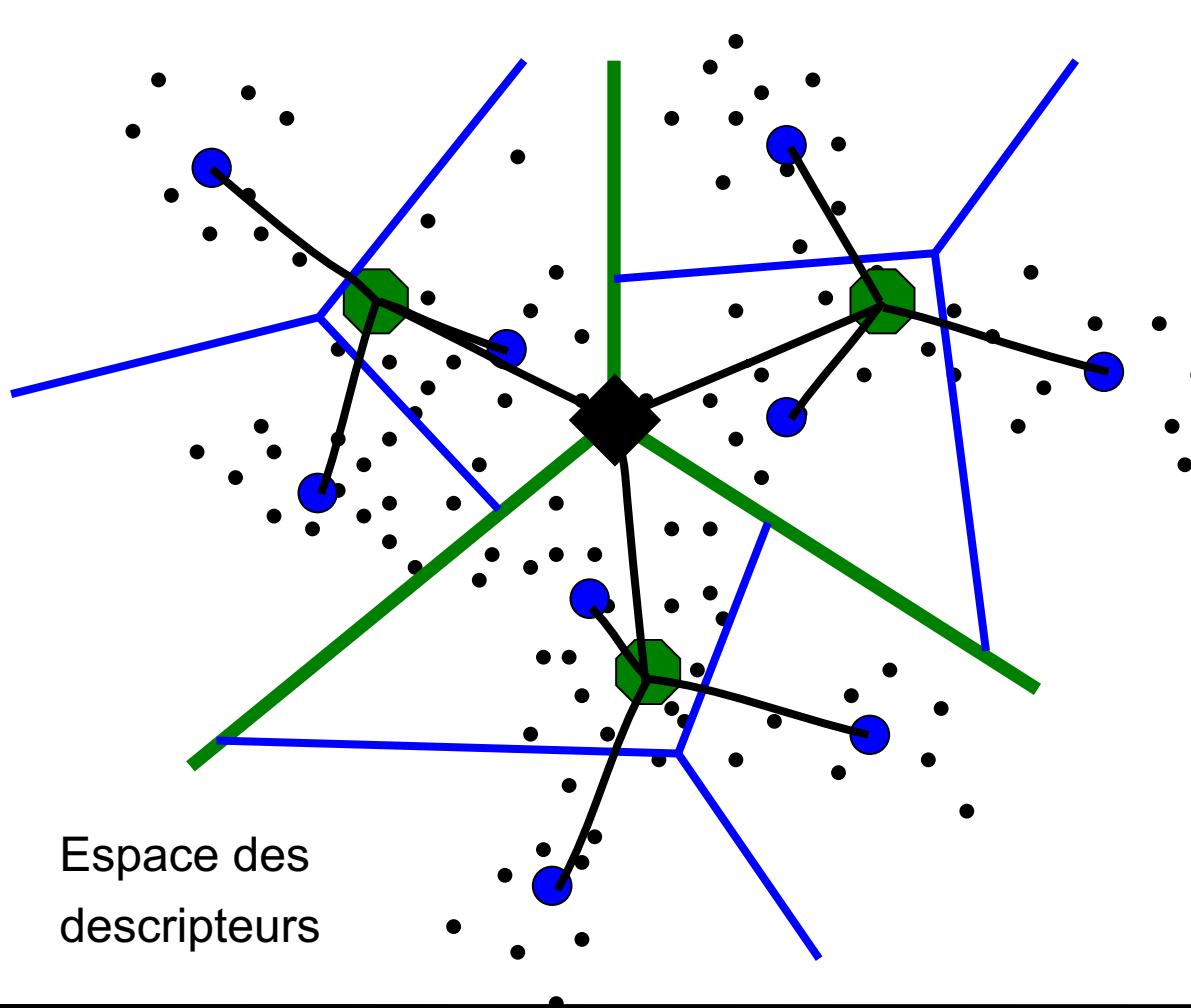
K-moyennes hiérarchique

- Approche à base d'arbre différente de kd-tree
- Partitionnement des descripteurs: k-moyennes (k-means)



K-moyennes hiérarchique

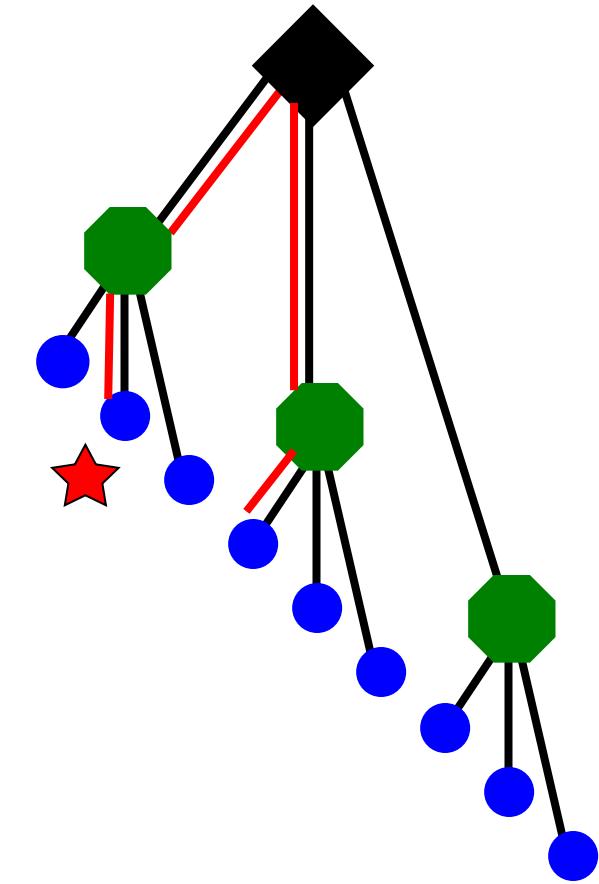
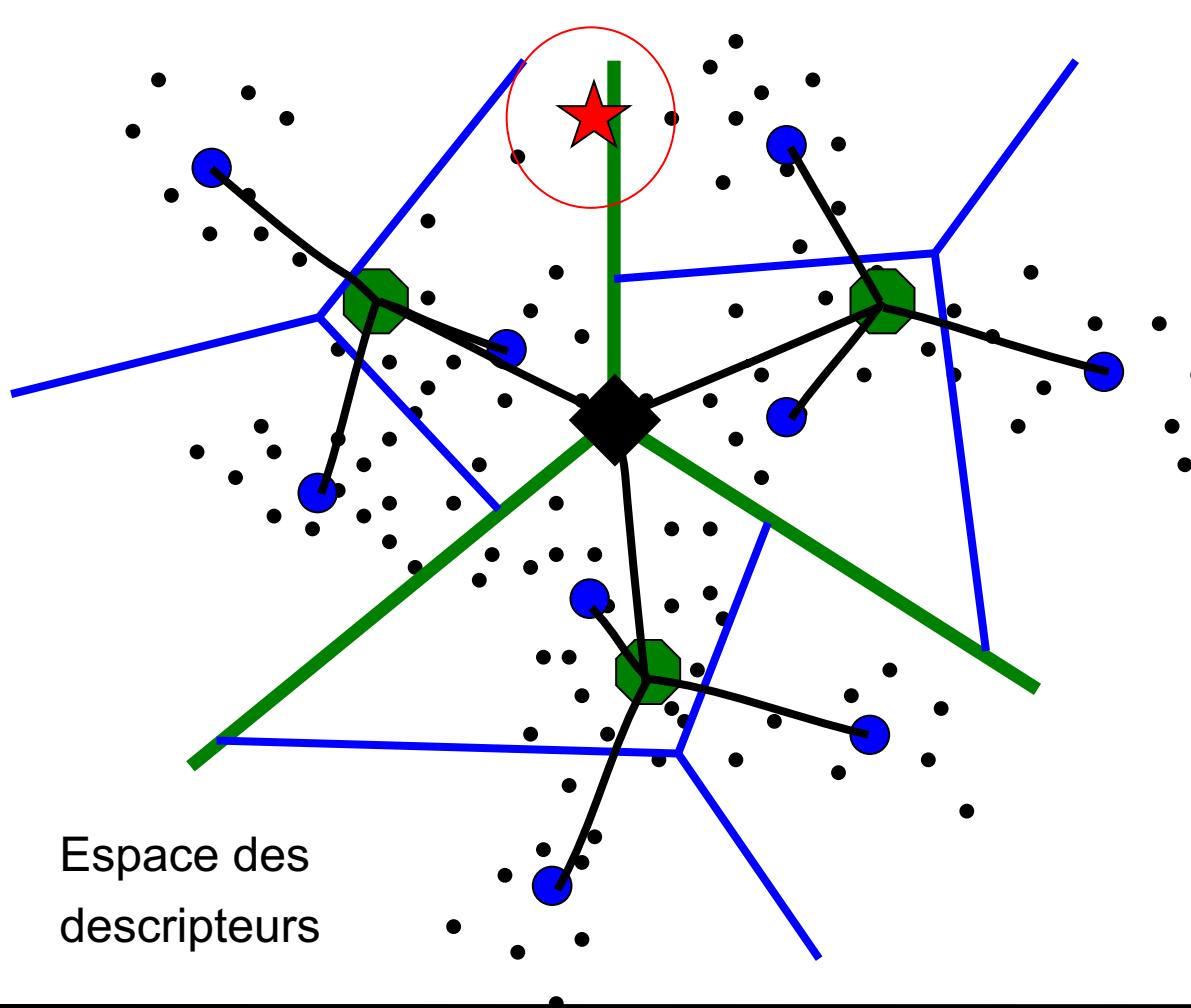
- Application récursive des k-moyennes (ex k=3)



[Nister & Stewenius, CVPR' 06]

K-moyennes hiérarchique

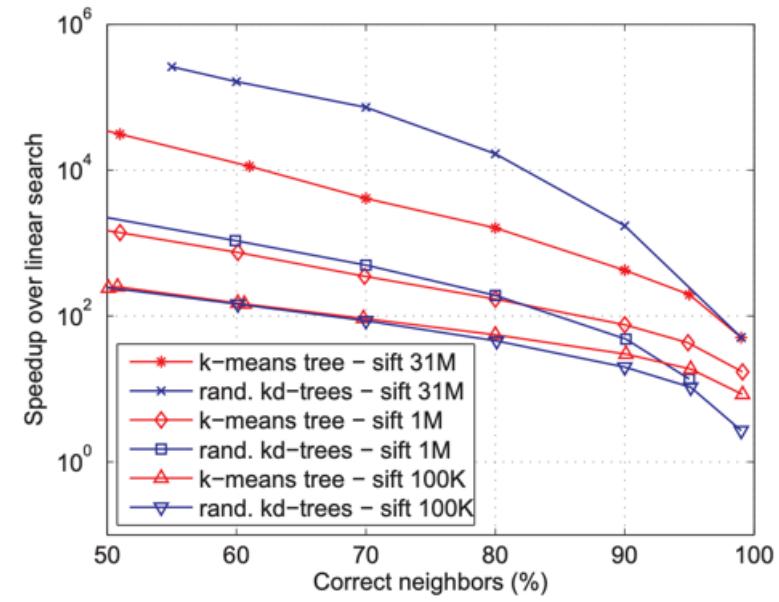
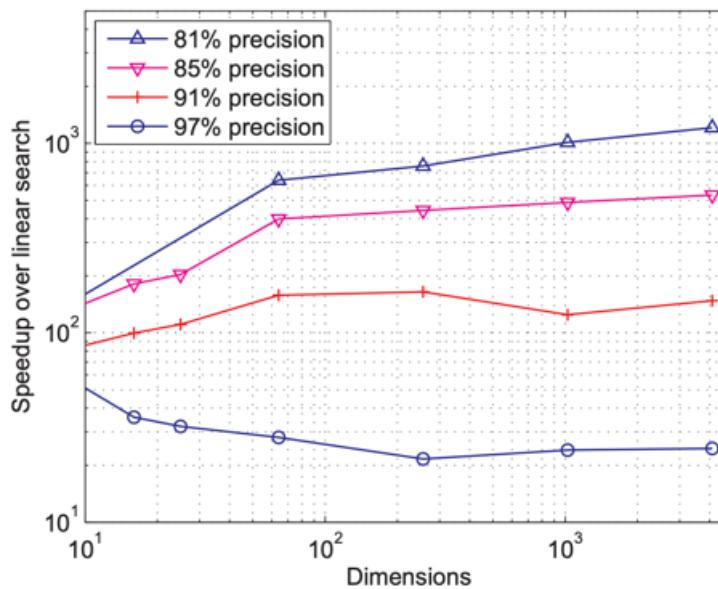
- Recherche similaire à kD-trees, peut exploiter BBF



[Nister & Stewenius, CVPR' 06]

Bibliothèque FLANN (Fast Library for Approximate Nearest Neighbor)

- « Marius Muja, David G. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration, 2009 »
- Implémente (multiple, randomized) kd-trees & k-moyennes hiérarchiques avec BBF
- Sélection automatique des paramètres pour une précision donnée



Modèle des sacs de mots visuels

- Mot visuel : groupe de descripteurs similaires

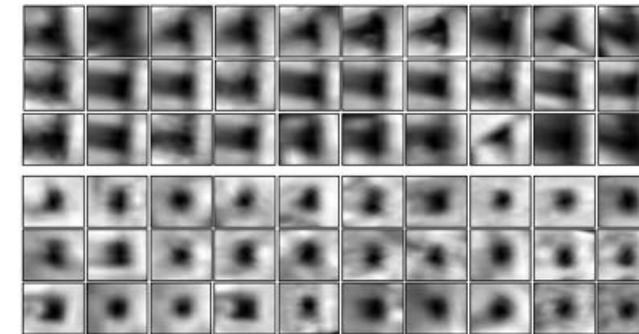
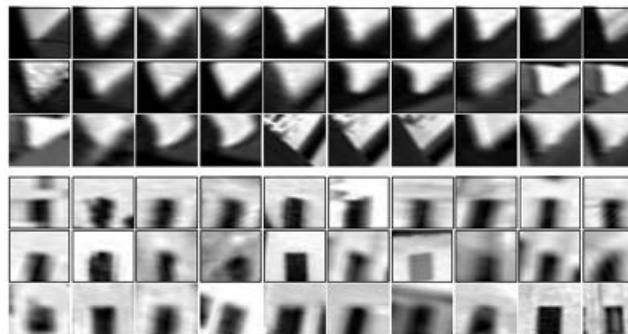
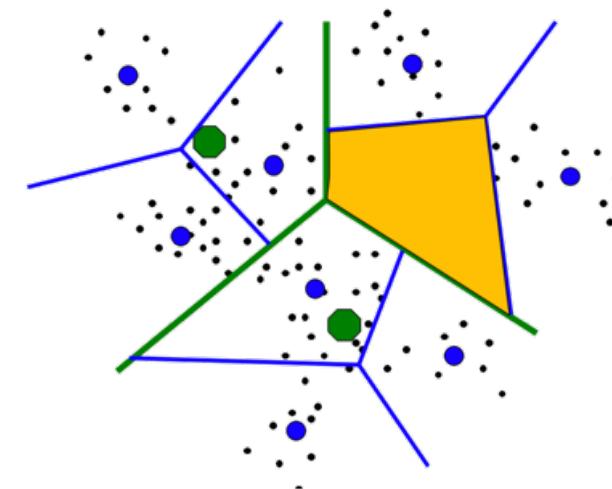


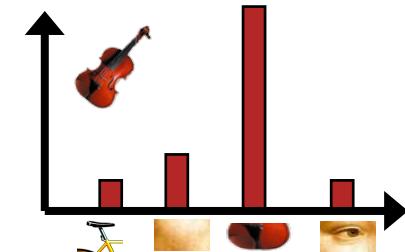
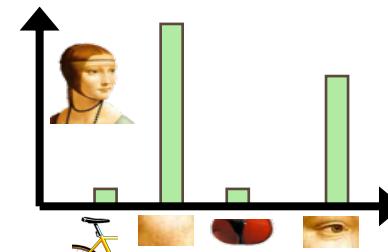
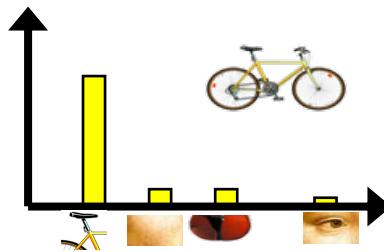
Figure from
Sivic & Zisserman, ICCV 2003

- L'ensemble des mots visuels forme un dictionnaire
 - Partitionnement de l'espace des descripteurs
 - Par exemple avec k-means hierarchique



Modèle des sacs de mots visuels

- Un objet est représenté par un ensemble de mot visuels
 - Oubli de l'information de position dans l'image
 - Image -> histogramme d'occurrence des mots



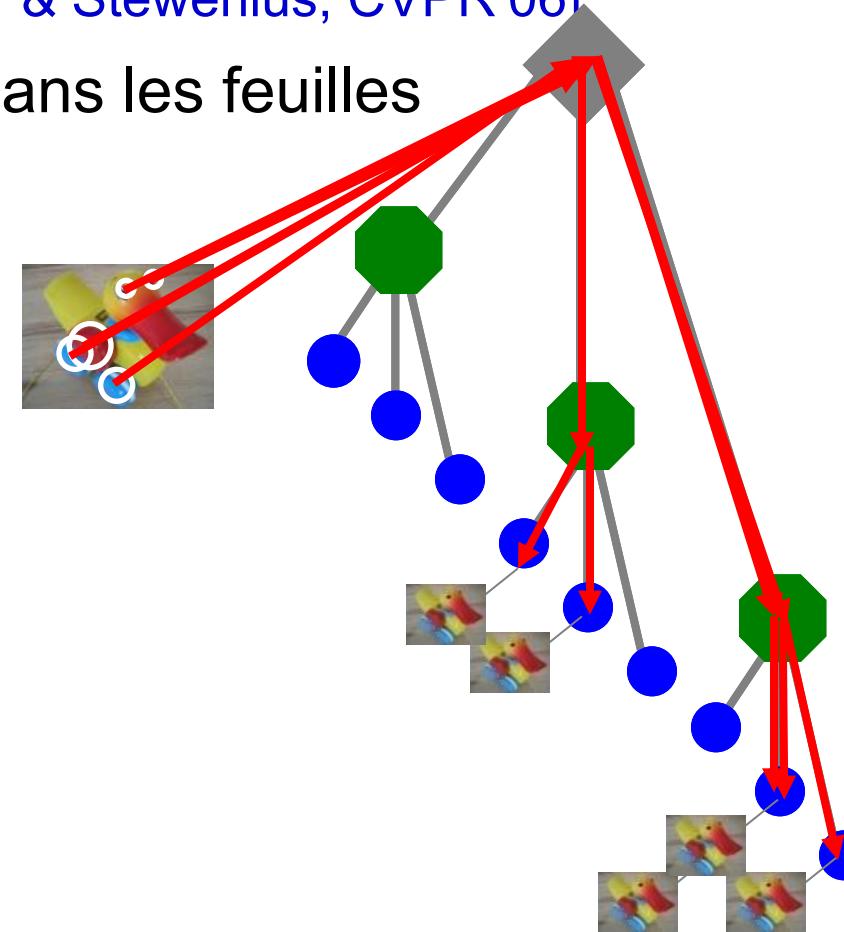
- Indexation des images via ‘index inversé’



Mot	Images
1	→ 5, 10, ...
2	→ 10, ...
...	...

Vocabulary tree [Nister & Stewenius, CVPR'06]

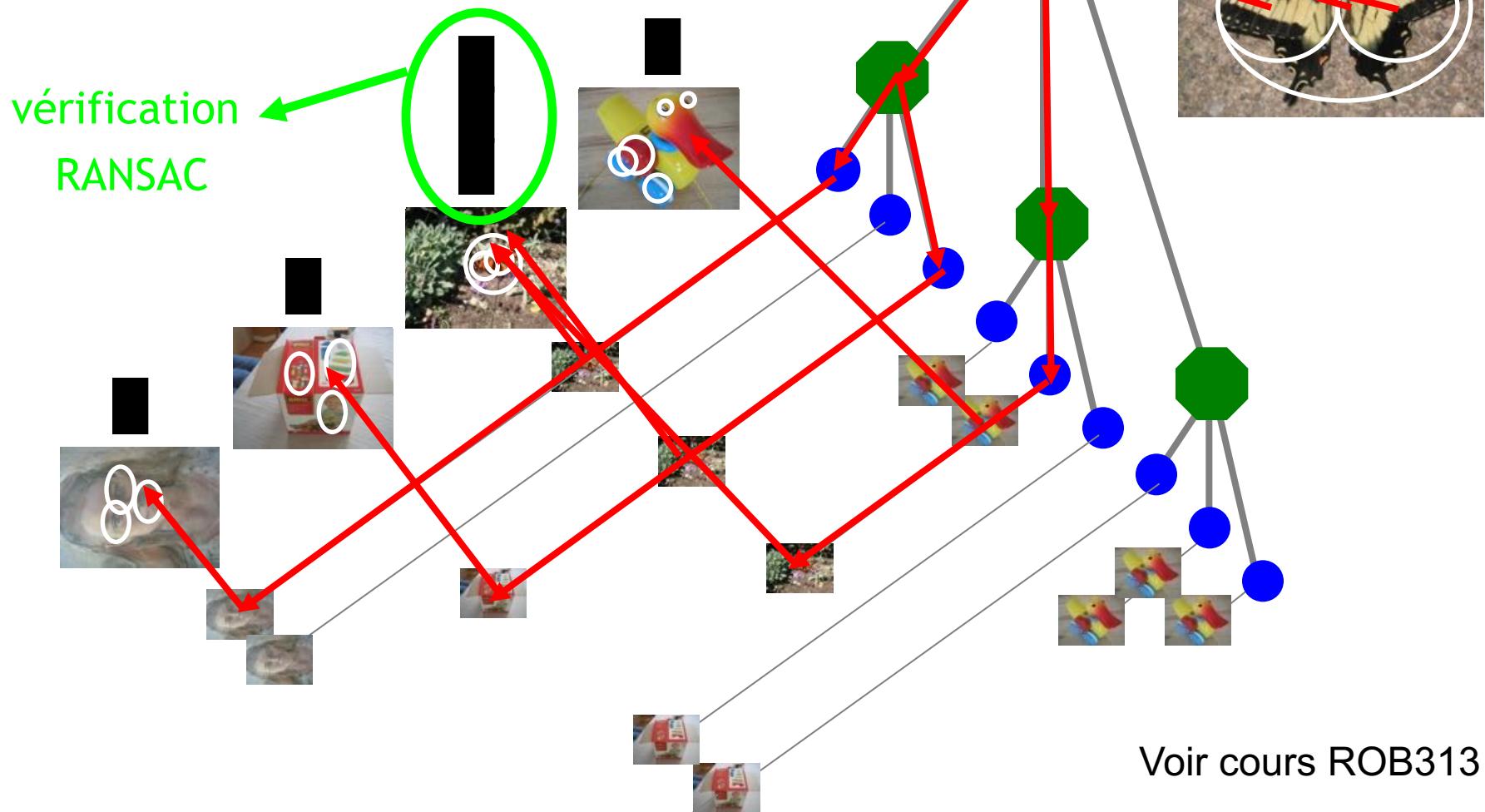
- Images indexées dans les feuilles



Slide credit: David Nister

Vocabulary tree

- Recherche par vote



Questions ?

TP python / OpenCV

