Passage à l'échelle des méthodes de recherche sémantique dans les grandes bases d'images

David GORISSE

Thèse encadrée par

Frédéric PRECIOSO
Matthieu CORD
Sylvie PHILIPP-FOLIGUET

20 décembre 2010

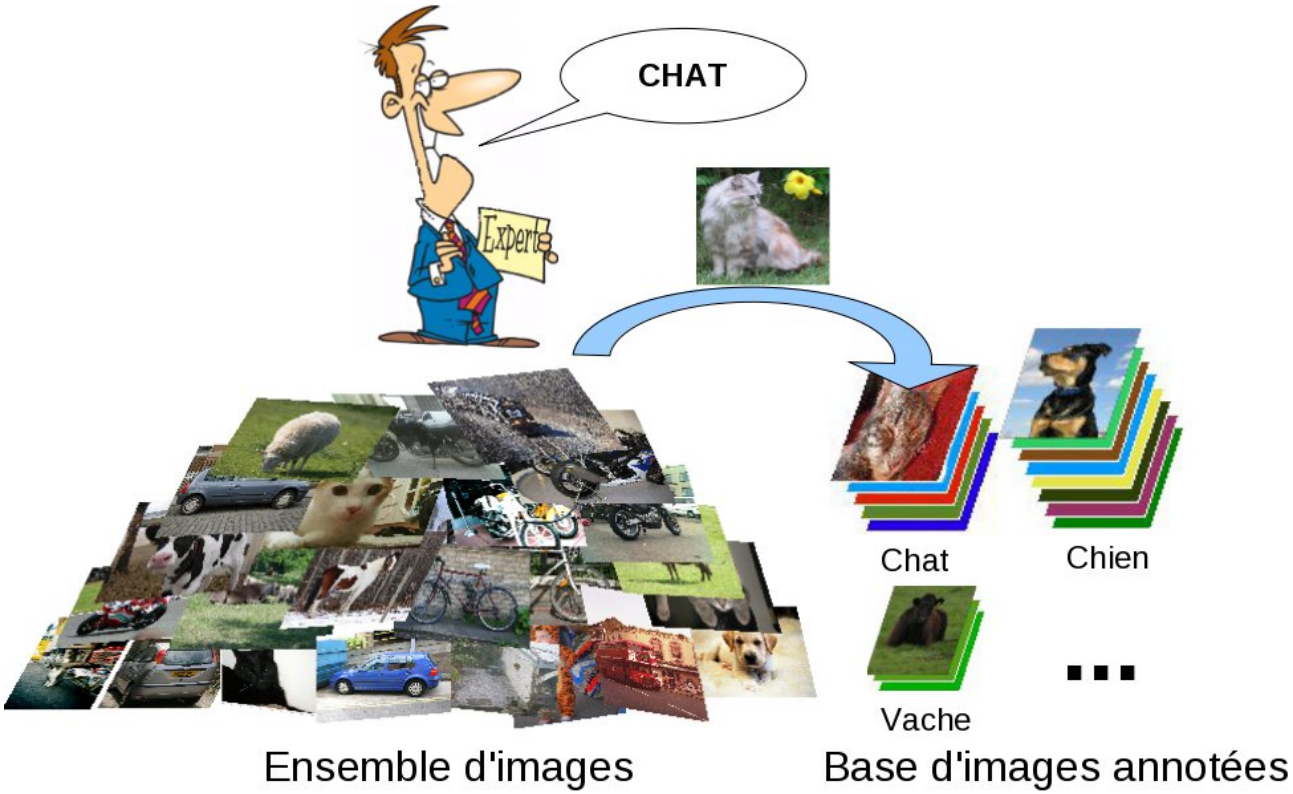
Introduction : Recherche d'information visuelle

Comment retrouver efficacement les images d'intérêt parmi un grand ensemble d'images ?

Introduction : Recherche d'information visuelle

Moteurs de recherche images basés sur l'information contextuelle :

1. Annotation manuelle de la base
2. Exploitation de métadonnées



Introduction : Recherche d'information visuelle

Moteurs de recherche images basés sur l'information contextuelle :

1. Annotation manuelle de la base
2. Exploitation de métadonnées

$\Rightarrow$ Requête par mots clés

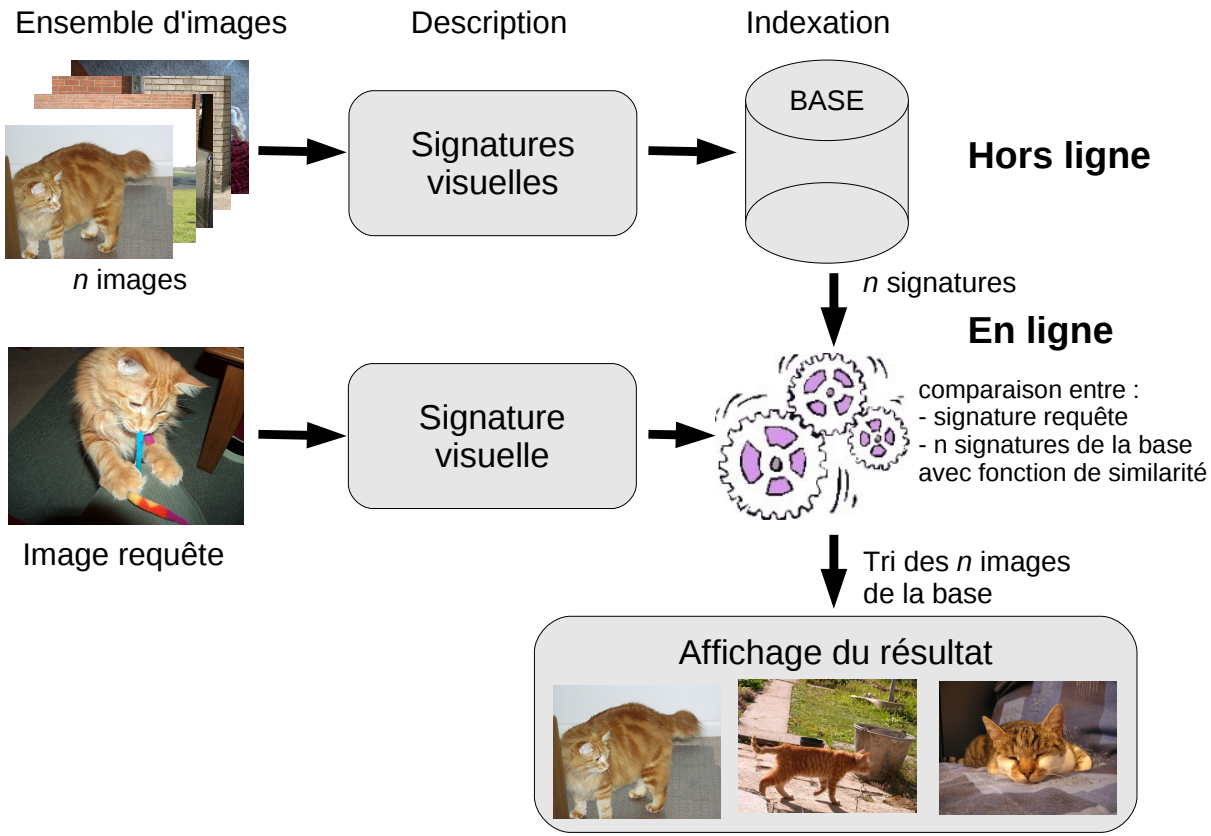
Introduction : Recherche d'information visuelle

Moteurs de recherche images basés sur l'information contextuelle :

1. Annotation manuelle de la base $\Rightarrow$ figé / peu évolutif
2. Exploitation métadonnées $\Rightarrow$ fiabilité ?

$\Rightarrow$ Requête par mots clés $\Rightarrow$ polysémie / ambiguïté

Introduction : Système de recherche d'images par le contenu (CBIR)



Etapes clés :

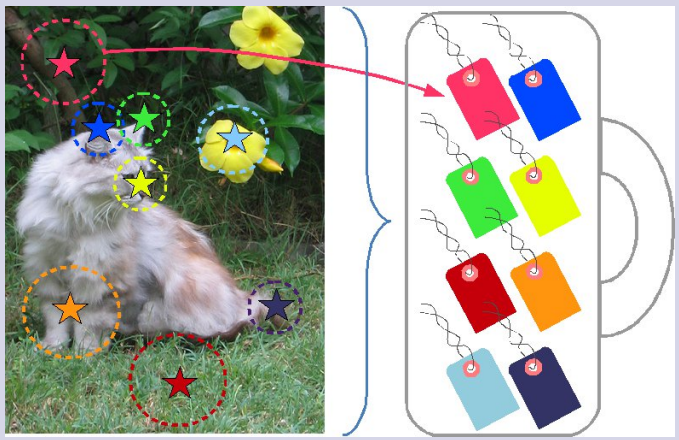
1. Signatures visuelles
2. Fonction de similarité

Introduction : Signatures

Représentation des images

- Extraction des primitives visuelles : pixels, régions, points d'intérêt
- Description : couleurs, textures, SIFT, ...

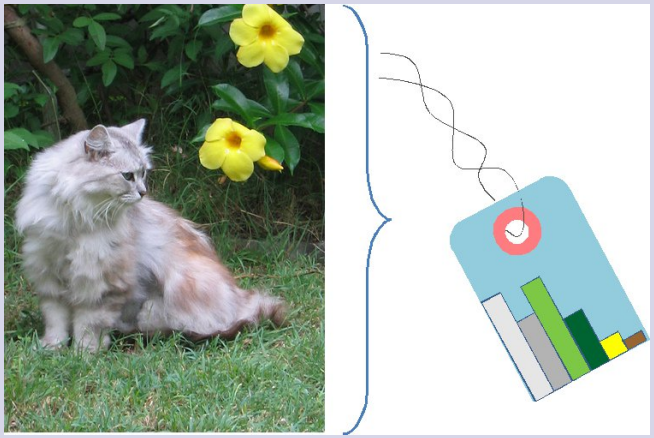
Bag of Features (BoF)



I_j $B_j = \{\mathbf{b}_{sj}\} \in I_j$

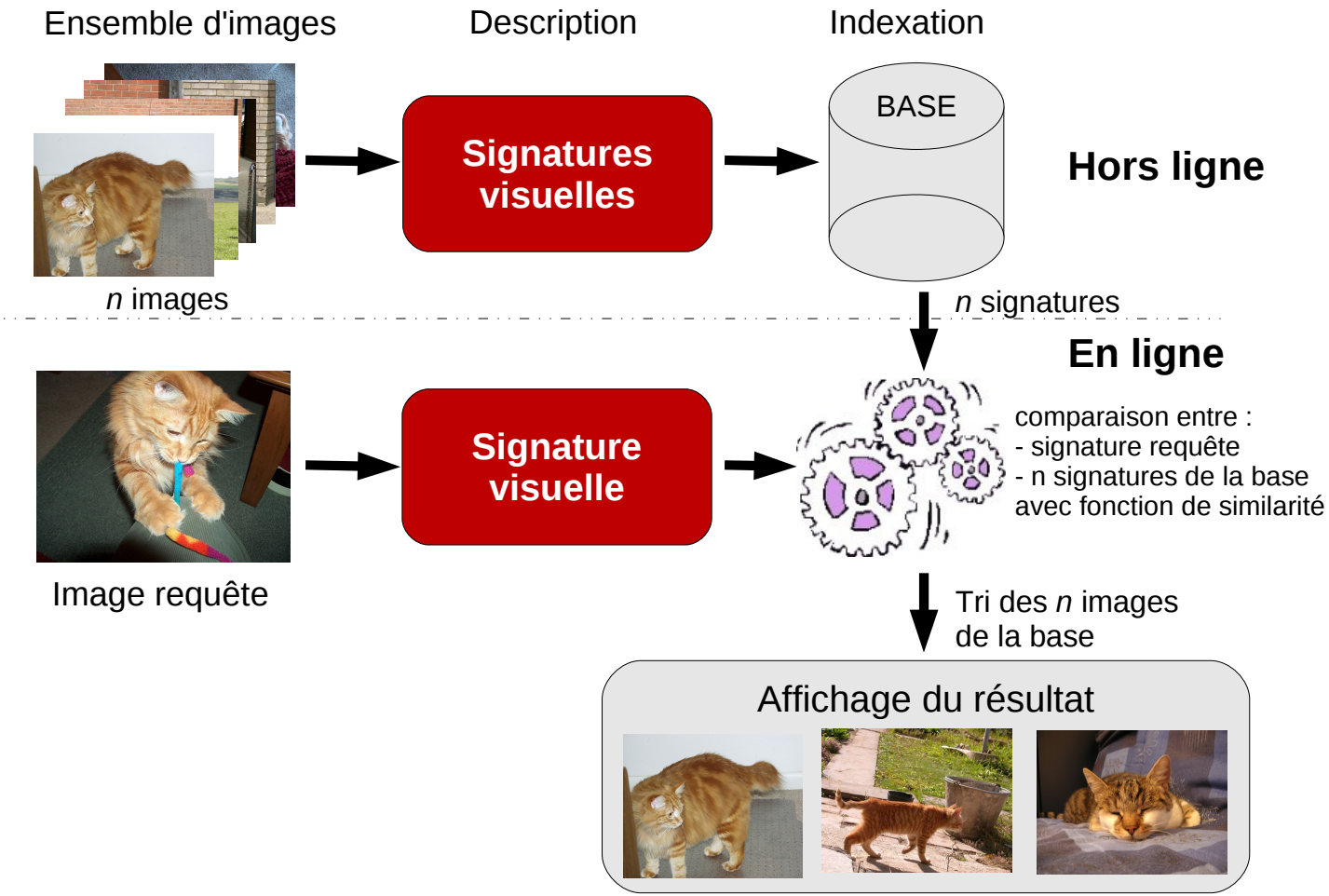
Bag of Word (BoW)

- Dictionnaire \ Histogramme

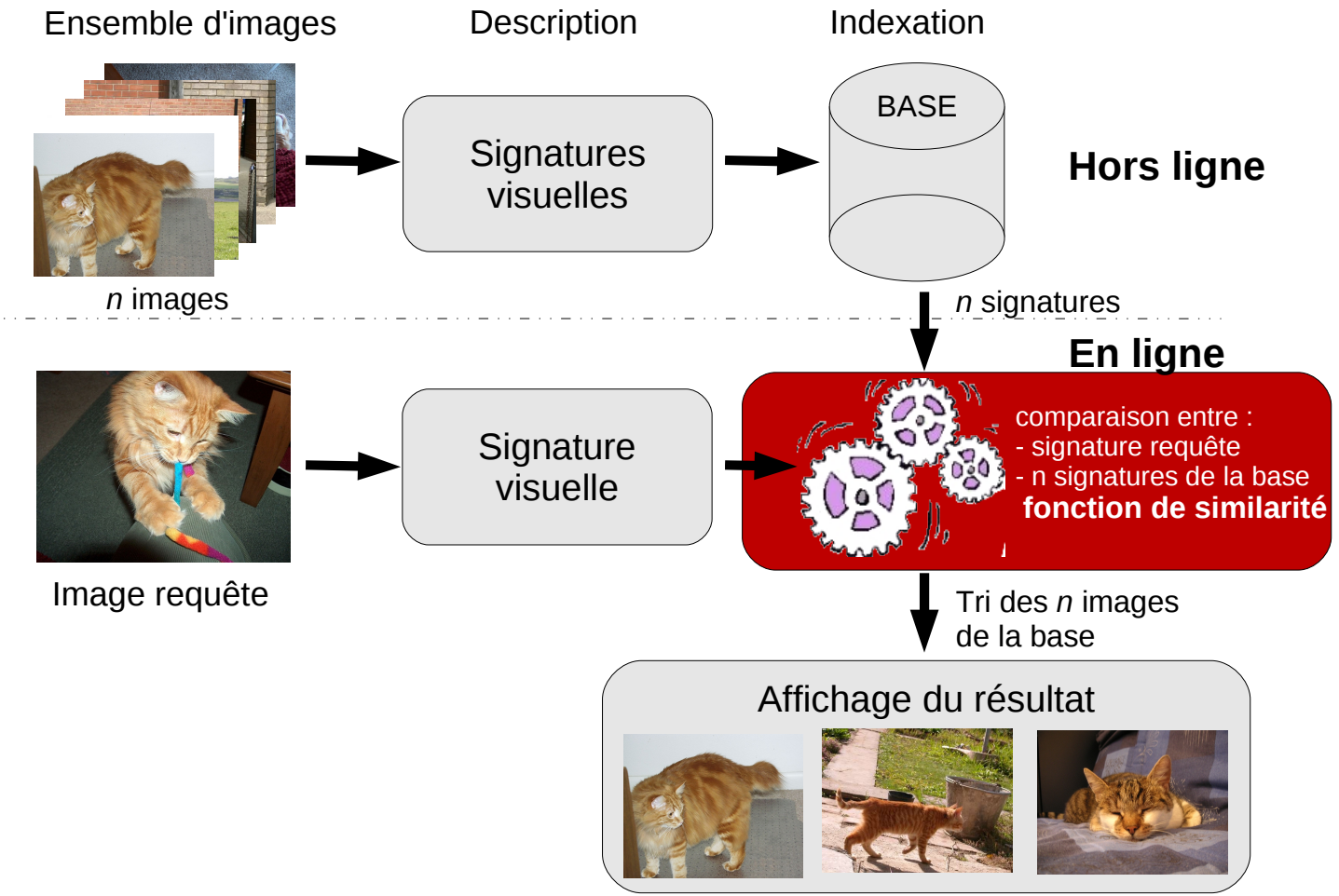


I_j $H_j = |w_0|, \ldots, |w_d| \in I_j$

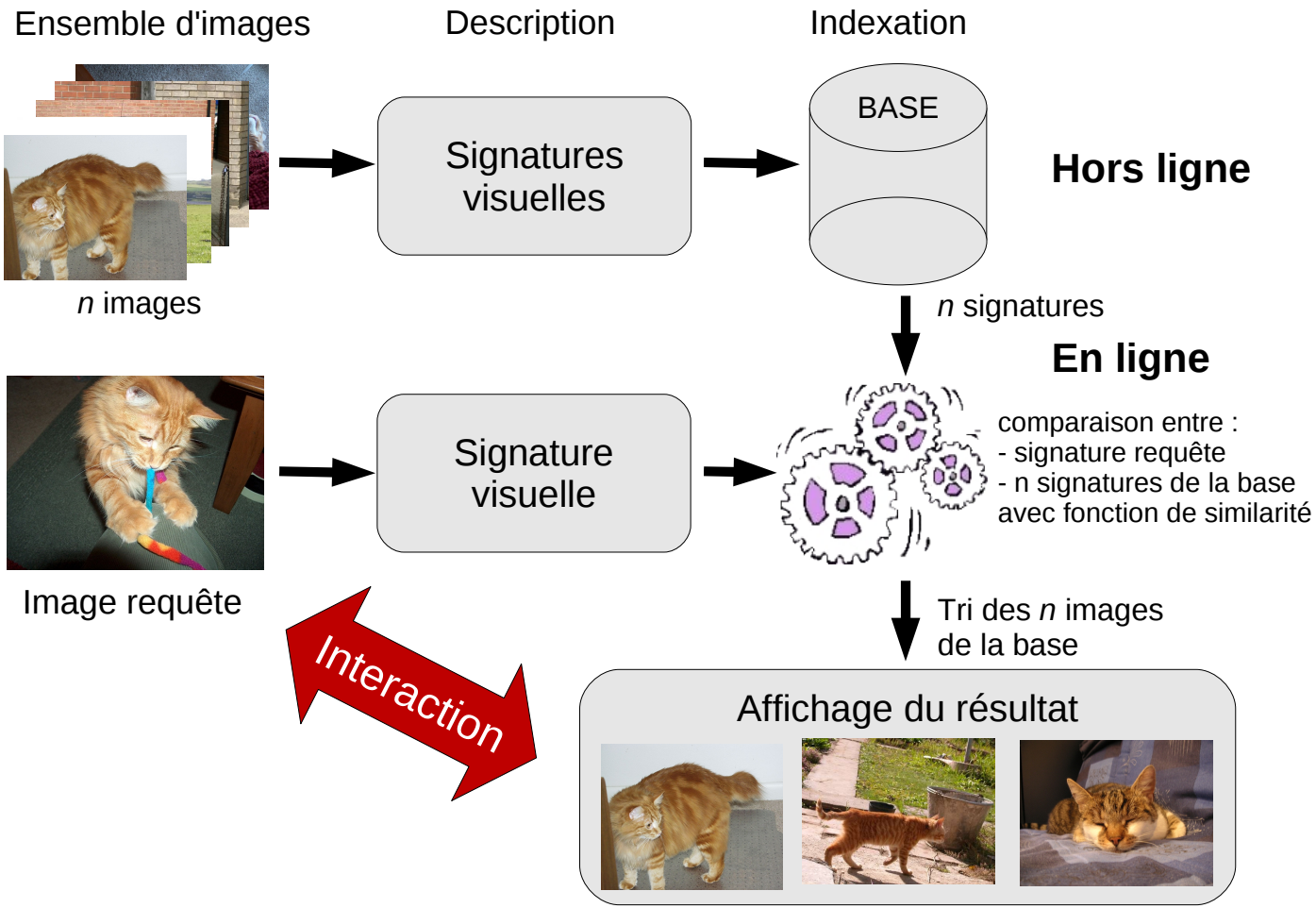
Introduction : Les problématiques en CBIR



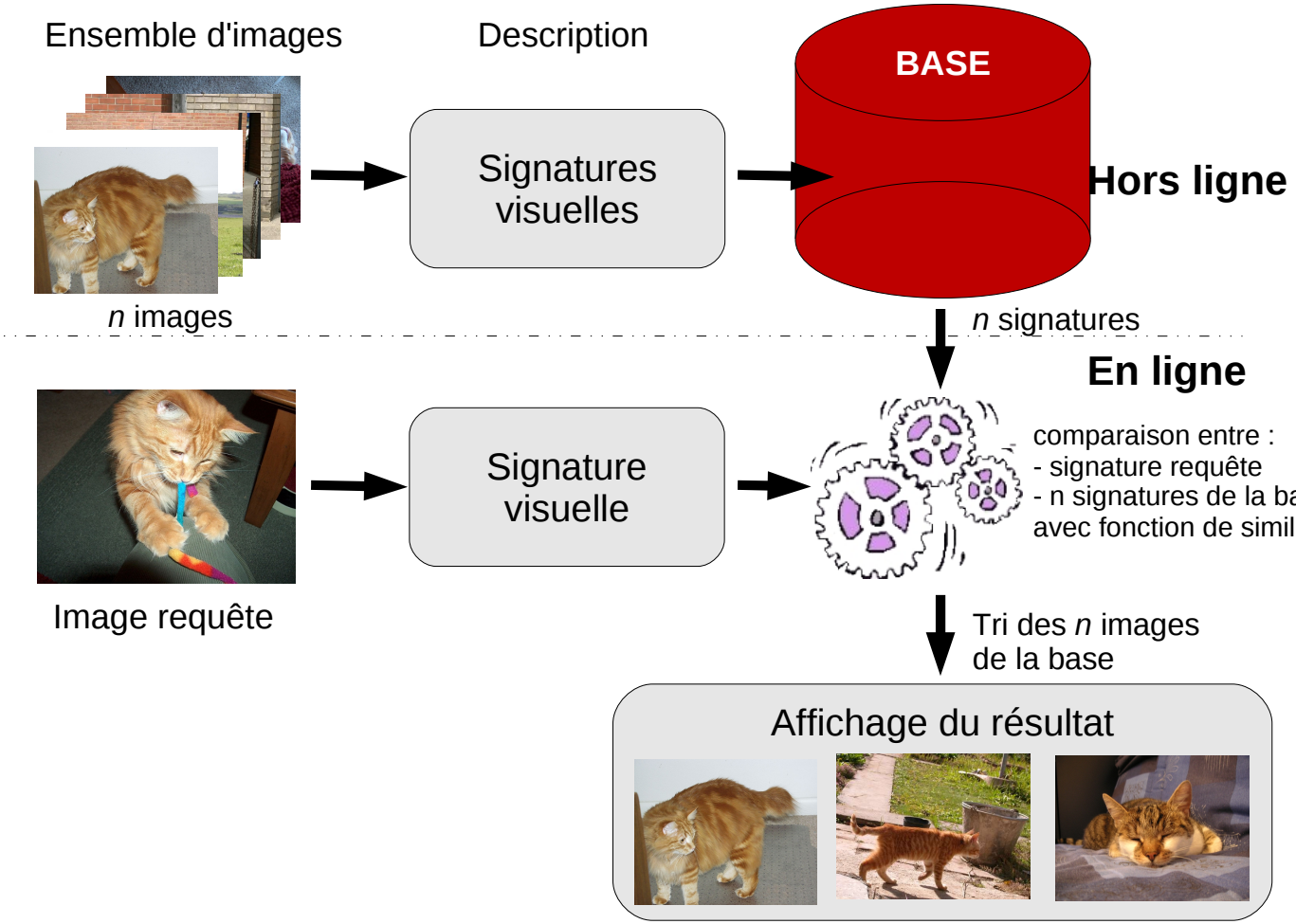
Introduction : Les problématiques en CBIR



Introduction : Les problématiques en CBIR



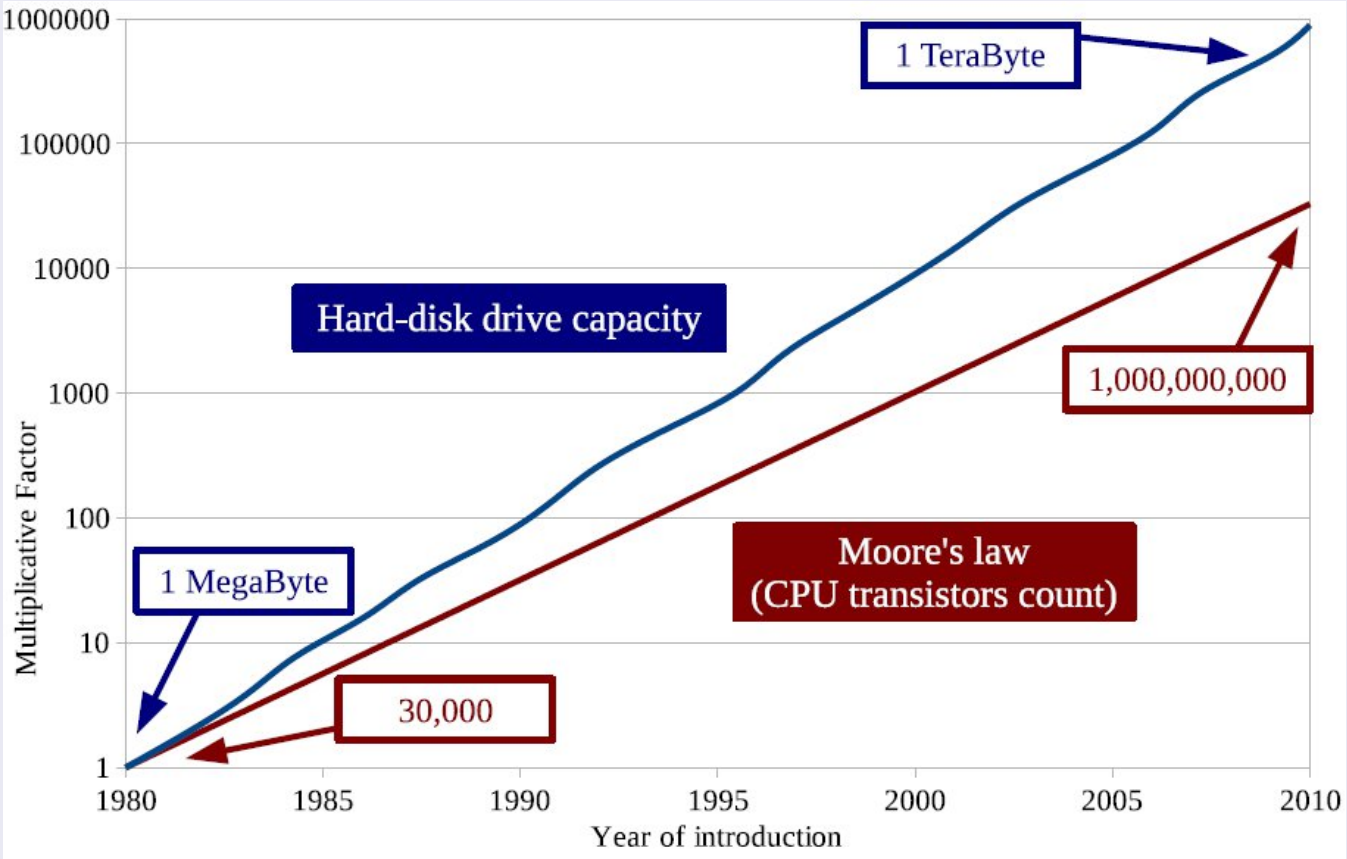
Introduction : Les problématiques en CBIR



Notre problématique : Complexité de la recherche

Comparaison entre image requête et les n images de la base : $O(n)$

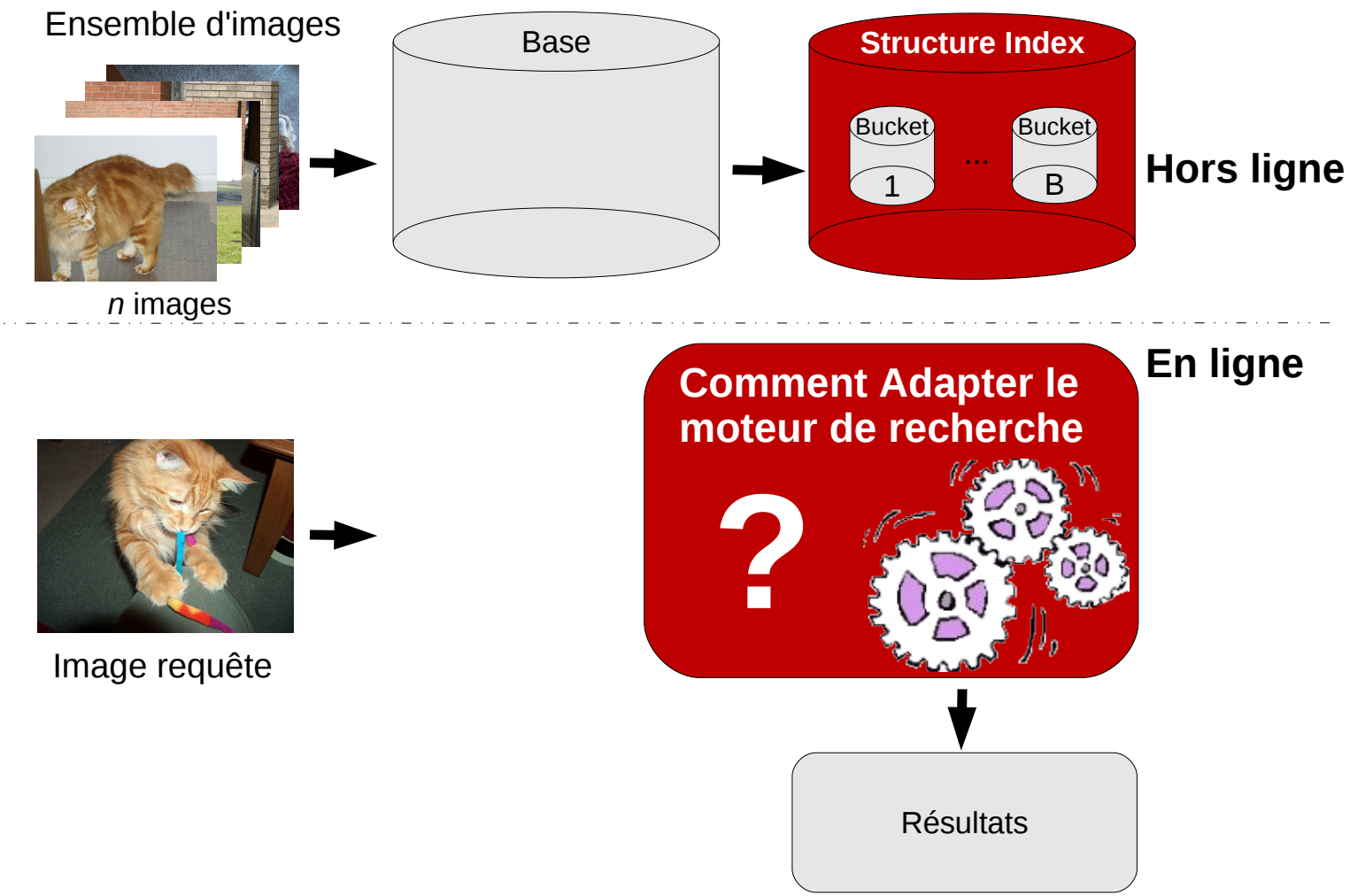
Introduction : De plus en plus d'images



Evolution vitesse des processeurs VS capacité de stockage

Besoin de réduire la complexité des moteurs de recherche !

Introduction : Utilisation de structures d'index



Problématique

Comment adapter les structures aux systèmes de recherche ?

Plan

Plan

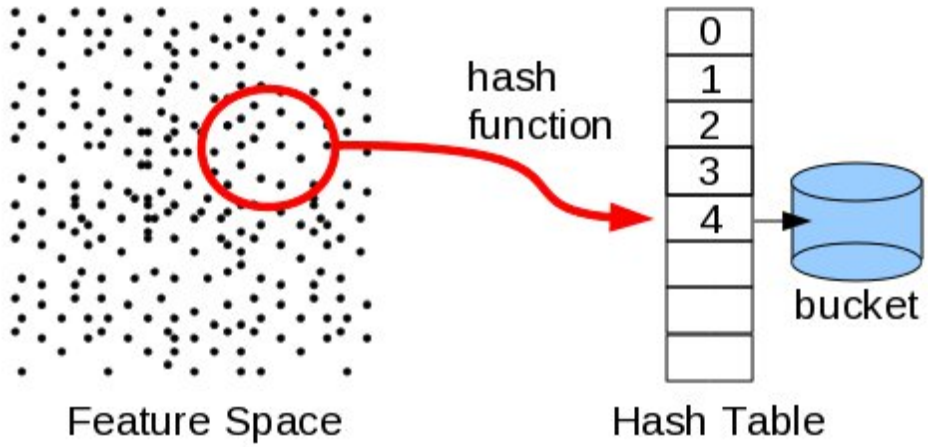
Locality Sensitive Hashing (LSH) [Broder97],[Indyk98]

Structures d'index

Arbre (KD-Tree), Projections (Space-Filing Curves), Métrique (M-Tree), Approximation des données (VA-Files), Hachage (**LSH**), ...

Vocabulaires

- Division de la base en Buckets
- Buckets $\rightarrow$ table de hachage
- Accès Bucket via un clé (haché)
- Clé $\rightarrow$ fonction de hachage h
- Collision : $h(\mathbf{x}) = h(\mathbf{y})$



Contrainte

- p_1 : probabilité de « bonne » collision (vrais positifs)
- p_2 : probabilité de « mauvaise » collision (faux positifs)
- LSH : $p_1 > p_2$

E2-LSH : LSH pour distance l_2 [Datar04]

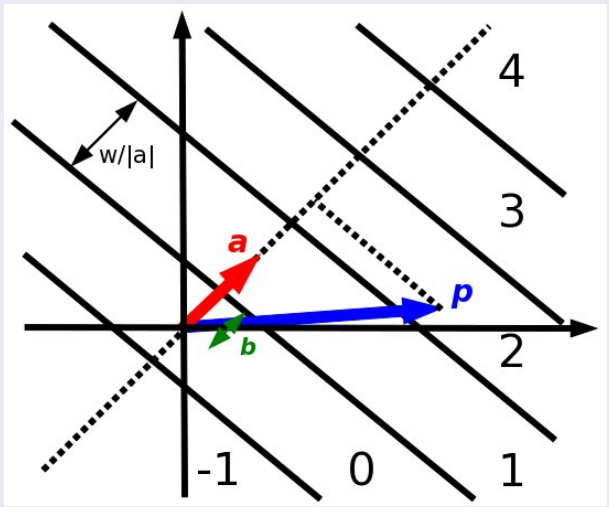
Definition

Réaliser des recherches rapides selon la distance l_2 :

- Fonction de hachage $h_{\mathbf{a},b}$ basée sur un ensemble de projections aléatoires :

$$h_{\mathbf{a},b}(\mathbf{p}) = \lfloor \frac{\mathbf{a}.\mathbf{p} + b}{w} \rfloor$$

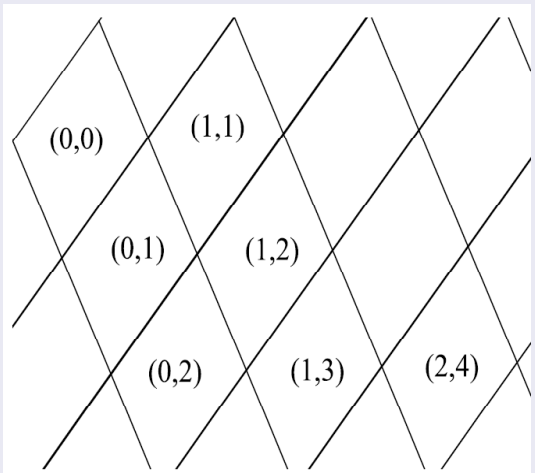
  - $\mathbf{a}$: un vecteur aléatoire
  - b : une valeur aléatoire dans $[0, W[$
  - W : spécifie la largeur d'une « tranche »



- Pour augmenter le partitionnement des données : concaténation de M fonctions de hachage.

- Pour réduire les problèmes de bords :
  - Utilisation de L tables de hachage.

- Extensions : MP-LSH [Lv2007], Leech Lattice [Andoni06], E8 Lattice [Jegou08], ...



Plan

Passage à l'échelle de la détection de copies

BUT : détection de copies proches

Retrouver un même objet dans un environnement différent (ou pris sous différents angles de vues)

Image requête

Images Cibles

Images Négatives

Détection de copies proches : comparaison entre images [SoA]

Approche par vote [Lowe04]

- Bag of Features (SIFT) : Dénombrement des appariements

Approche dictionnaire [Zisserman03]

- Bag of Word (mots SIFT) : Distances l_1

Détection de copies proches : comparaison entre images [SoA]

Approche par vote [Lowe04]

- Bag of Features (SIFT) : Dénombrement des appariements

$\Rightarrow$ Pas de similarité (ni de dissimilarité) !

Approche dictionnaire [Zisserman03]

- Bag of Word (mots SIFT) : Distances l_1

$\Rightarrow$ Suffisamment discriminant ?

Détection de copies proches : comparaison entre images [SoA]

Approche par vote [Lowe04]

- Bag of Features (SIFT) : Dénombrement des appariements

$\Rightarrow$ Pas de similarité (ni de dissimilarité) !

Approche dictionnaire [Zisserman03]

- Bag of Word (mots SIFT) : Distances l_1

$\Rightarrow$ Suffisamment discriminant ?

Alternatives

- Bag of Features : Définir une fonction similarité sur BoF (Pyramid Matching [Grauman05])
- Bag of Word : Distribution plus fine (représentation de Fisher kernel [Perronnin07], VLAD [Jegou10])

Détection de copies proches : comparaison entre images [SoA]

Approche par vote [Lowe04]

- Bag of Features (SIFT) : Dénombrement des appariements

$\Rightarrow$ Pas de similarité (ni de dissimilarité) !

Approche dictionnaire [Zisserman03]

- Bag of Word (mots SIFT) : Distances l_1

$\Rightarrow$ Suffisamment discriminant ?

Alternatives

- Bag of Features : Noyaux sur Sacs

Notations :

$$\begin{aligned} \text{Soit } k : \Omega \times \Omega &\longrightarrow \mathbb{R} \\ X, Y &\longmapsto k(X, Y) \end{aligned}$$

- k est un Noyau *ssi* :

$$\exists \phi | \forall (X, Y), k(X, Y) = \langle \phi(X), \phi(Y) \rangle$$

- avec $\phi : \Omega \longrightarrow \mathcal{H}$ (espace de Hilbert)
- ϕ : explicite ou implicite

Noyaux usuels pour $\Omega \longrightarrow \mathbb{R}^d$:

- RBF : $k(\mathbf{x}, \mathbf{y}) = e^{-\frac{d(\mathbf{x},\mathbf{y})^2}{2\sigma^2}}$, l_2-RBF : avec $d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_2$

Définition : K_{conv}

$$K_{conv}(B_q, B_j) = \sum_{\mathbf{b}_{sj} \in B_j} \sum_{\mathbf{b}_{rq} \in B_q} k(\mathbf{b}_{sj}, \mathbf{b}_{rq})$$

si k est un Noyau $\Rightarrow$ K_{conv} est un Noyau (pour $p \in \mathbb{Z}^{+*}$)

Pour chaque couple $(\mathbf{b}_{rq}, \mathbf{b}_{sj})$: noyau mineur $k(\mathbf{b}_{rs}, \mathbf{b}_{sj})$

Image I_q représentée par : Sac B_q
r vecteurs $\mathbf{b}_{rq}$

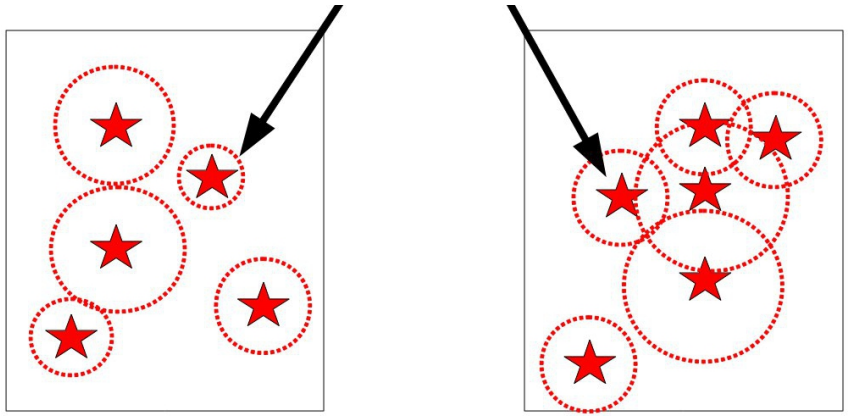


Image I_j représentée par : Sac B_j
s vecteurs $\mathbf{b}_{sj}$

Noyau sur sac : Complexité

Noyaux sur sacs : K_{conv}

Bons Résultats

MAIS Complexité très importante :

similarité entre 2 images $O(|B|^2)$, recherche dans une base : $O(n|B|^2)$

Diminution de la complexité :

- Pyramid Match Hashing [Grauman06]

$\Rightarrow$ bonne performance

- MAIS ne s'étend pas aux noyaux ou Φ n'est pas explicite

Si la classe des noyaux est différente (ex : K_{conv})

- Peut-on accélérer le calcul ?

Noyau sur sac : Complexité

Noyaux sur sacs : K_{conv}

Bons Résultats

MAIS Complexité très importante :

similarité entre 2 images $O(|B|^2)$, recherche dans une base : $O(n|B|^2)$

Diminution de la complexité :

- Pyramid Match Hashing [Grauman06]

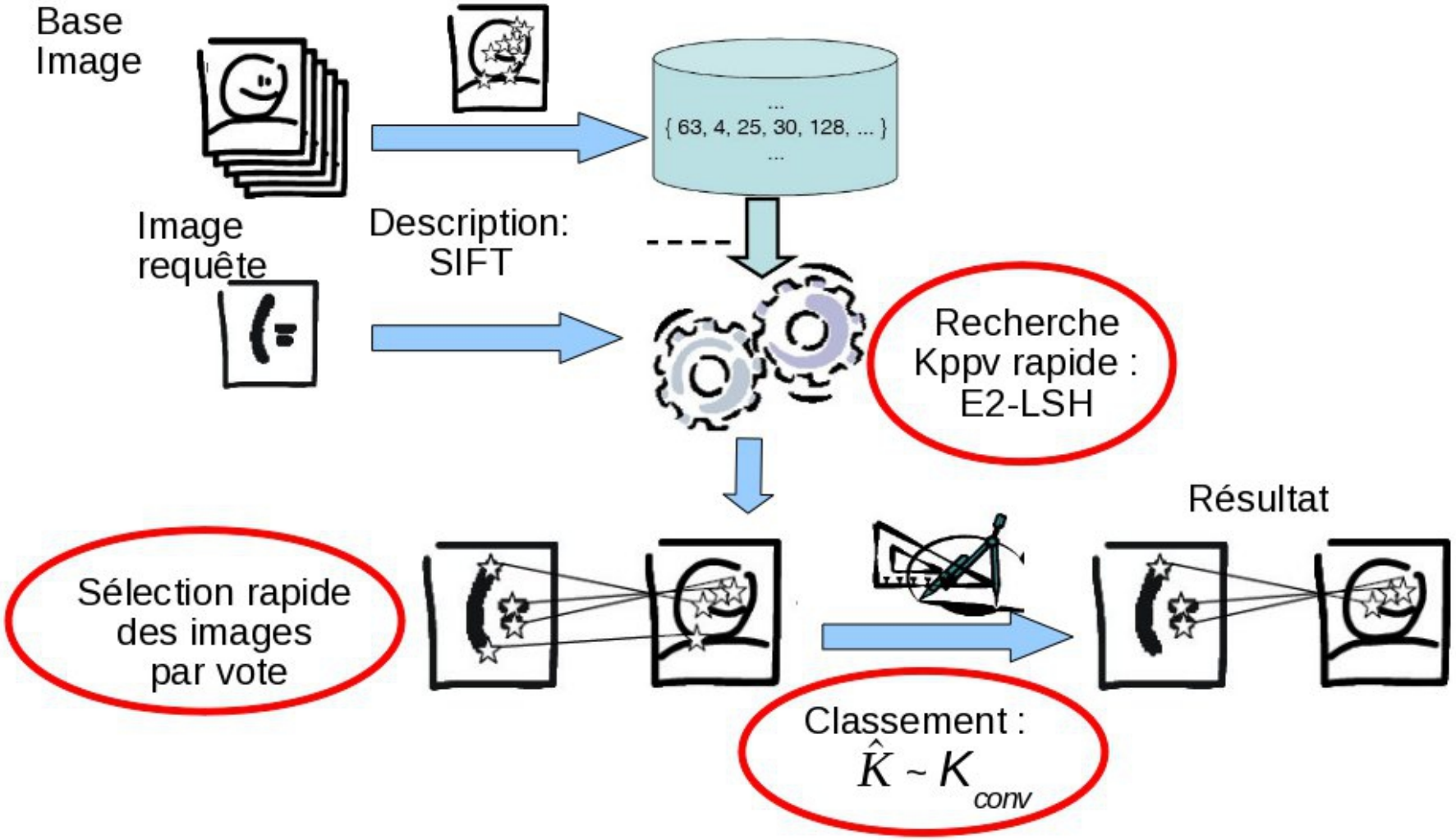
$\Rightarrow$ bonne performance

- MAIS ne s'étend pas aux noyaux ou Φ n'est pas explicite

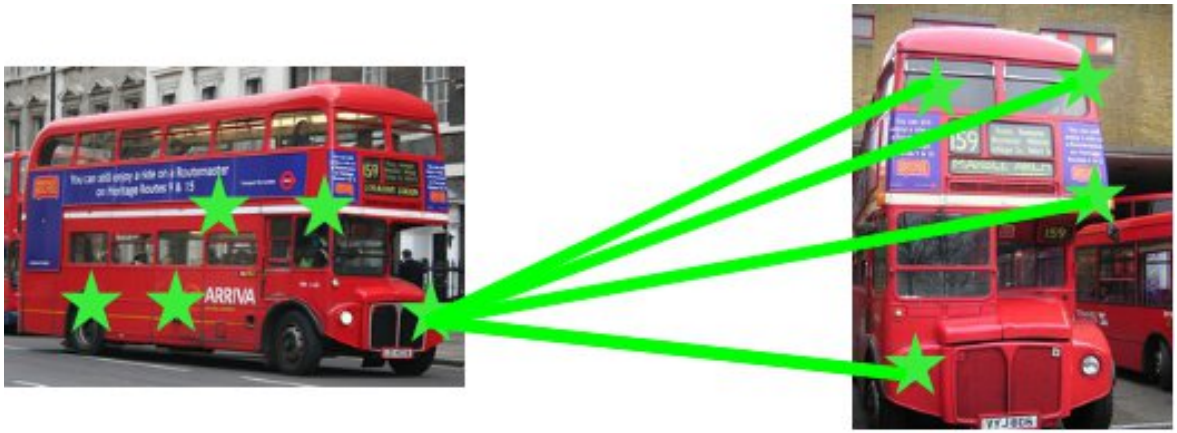
Si la classe des noyaux est différente (ex : K_{conv})

- Peut-on accélérer le calcul ?
- $UFast_K$: schéma d'optimisation inspiré de l'approche par vote.

$UFast_K$


Base
Image
...
{ 63, 4, 25, 30, 128, ... }
...
Image
requête
Description:
SIFT
Recherche
Kppv rapide :
E2-LSH
Résultat
Sélection rapide
des images
par vote
Classement :
$\hat{K} \sim K_{conv}$

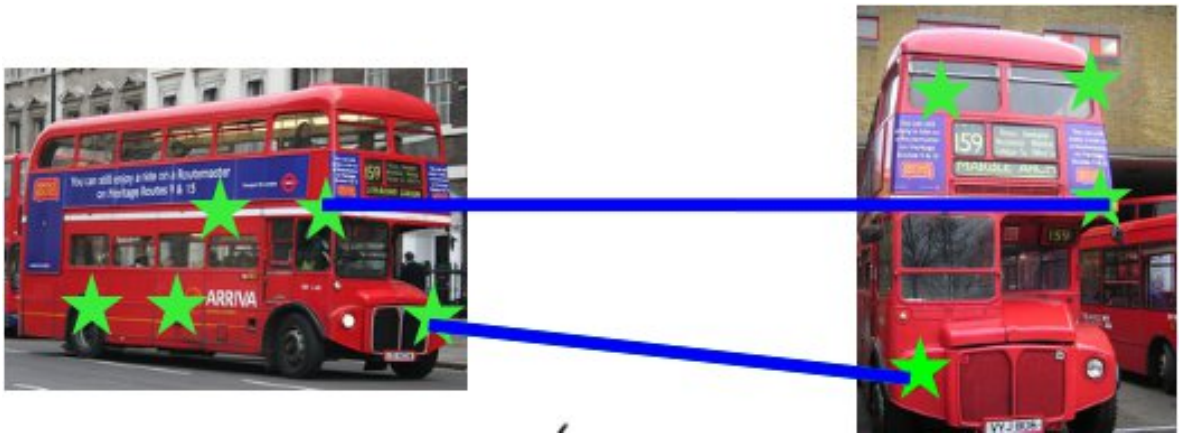
$UFast_K$: Approximation de K



- Calcul de $k(\mathbf{b}_{rq}, \mathbf{b}_{sj})$ pour :
  - $\forall \mathbf{b}_{rq} \in B_q$
  - $\forall \mathbf{b}_{sj} \in B_j$

$\Rightarrow O(|B_q||B_j|)$

Complexité trop importante !



- Calcul de $k(\mathbf{b}_{rq}, \mathbf{b}_{sj})$ uniquement pour :
  - couples $(\mathbf{b}_{rq}, \mathbf{b}_{sj})$ retrouvés via LSH

Réduction de Complexité

Approximation : Ultra Fast Kernel

Formalisme unifié

- $score_j = UFast_K(B_j, B_q) = \sum_{\mathbf{b}_{rq} \in B_q} \sum_{\mathbf{b}_{sj} \in B_j} k(\mathbf{b}_{rq}, \mathbf{b}_{sj}) f_{LSH}(\mathbf{b}_{rq}, \mathbf{b}_{sj})$

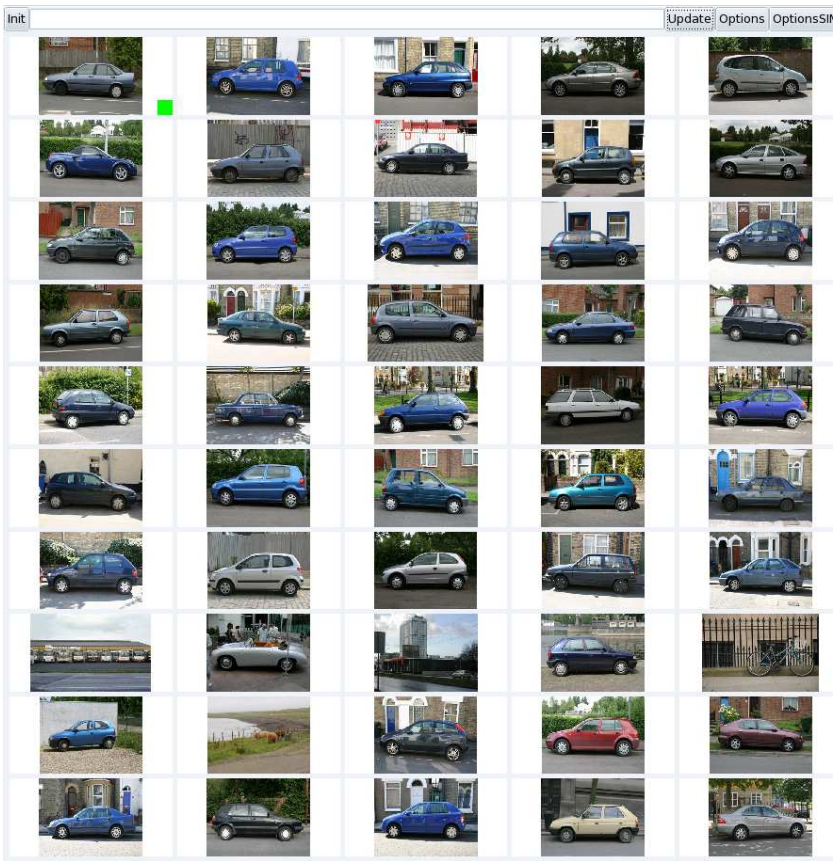
avec $f_{LSH}(\mathbf{b}_{sj}, \mathbf{b}_{rq}) = \mathbb{1}_{LSH(\mathbf{b}_{rq}) = LSH(\mathbf{b}_{sj})}$

- Vote : $score_j = \sum \sum \mathbb{1}_{KNN(\mathbf{b}_{rq}, \mathbf{b}_{sj})}$
- BoW : $score_j = \sum \sum \mathbb{1}_{Q(\mathbf{b}_{rq}) = Q(\mathbf{b}_{sj})}$
- K_{conv}, VLAD

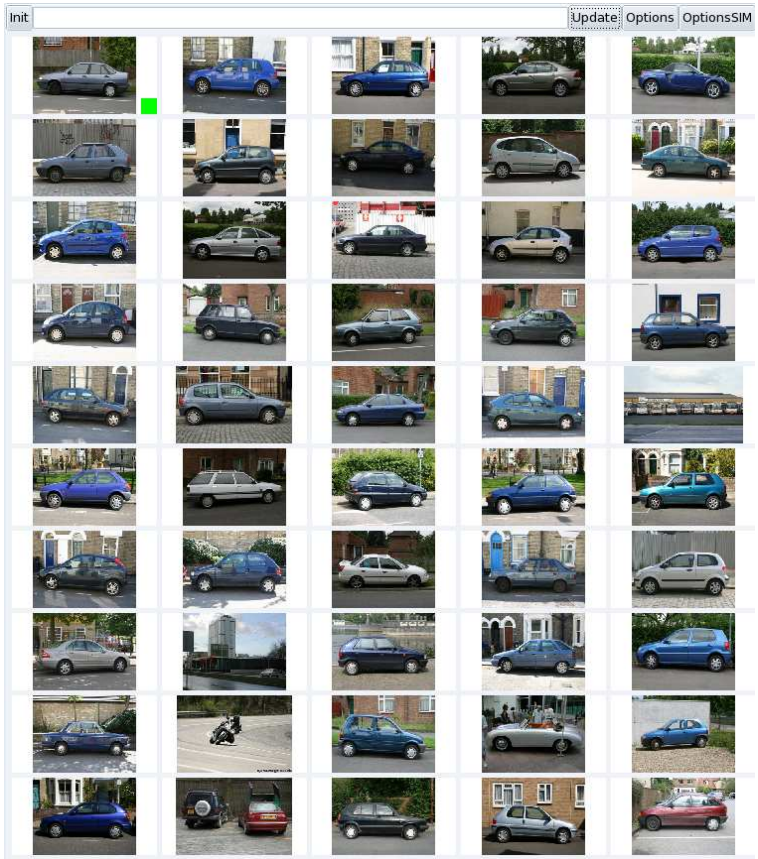
} [Jegou08]

Exemple

VOC2006 : 5 300 image, 260 PoI / image
1,4 Millions SIFT couleur (384 dimensions)



Recherche exacte : 6 minutes



$UFast_K$: 2,4 secondes

Evaluation

- 2 Bases :
  - INRIA Holiday = 1 491 images (dont 500 requêtes)
  - Holiday + Flickr = 100 000
- Index :
  - Descripteur SIFT (128 dimensions)
  - BoF (4 000 SIFT par image)

INRIA Holiday = 4,5 millions de descripteurs, Holiday + Flickr = 208 millions de descripteurs

- Protocole : MAP
  - 500 images requêtes (INRIA Holiday)
  - INRIA Holiday : verité terrain
  - Flickr : distracteurs

Comparison avec les méthodes de l'état de l'art

INRIA Holidays

Méthode	paramètres	MAP
BoW	$1K$ mots	41.4
	$20K$ mots	44.6
	$200K$ mots	54.9
VLAD	16 mots, taille du vecteur=2048	49.6
	64 mots, taille du vecteur=8192	52.6
$UFast_K$	$2NN\ M = 24\ L = 4\ T = 50$	76

$\Rightarrow$ Amélioration de la recherche en conservant une information plus fine

Résultats du passage à l'échelle

Base Holidays + Flickr (100K)

- MAP de 42
- 31 Go de RAM
- 40 secondes
- KNN pour 1 SIFT de l'image requête : 16 ms

Avec réduction du nombre de SIFT de l'image requête : 4 000 $\Rightarrow$ 200 SIFT

Temps de recherche de 3,2 secondes

Plan

Extension aux très grandes bases

1 image = 1 descripteur

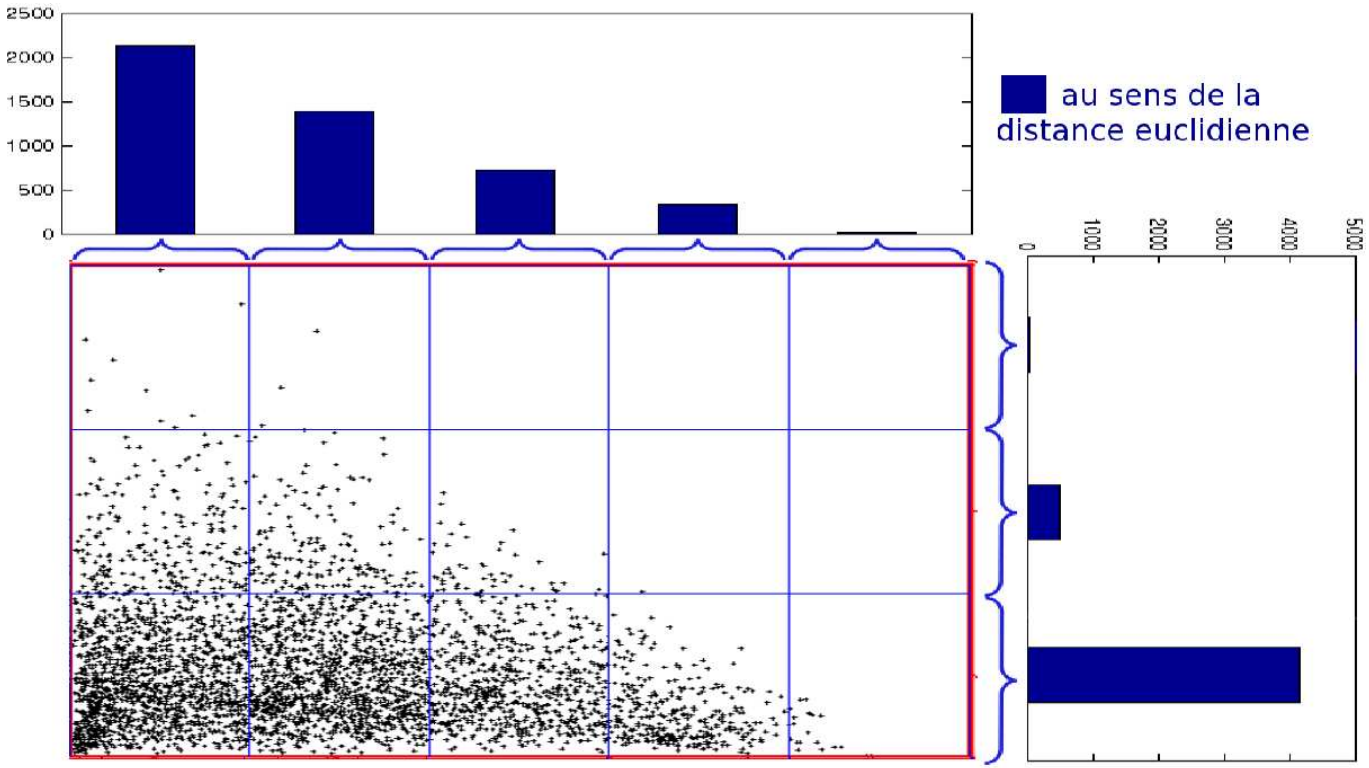
- Trés grandes Bases $>$ 10M images :
  - BoF plus possible $\Rightarrow$ BoW
- BoW : similarités basées distance χ^2 très compétitives en recherche d'images [Chapelle99, ...]
- Exemple : χ^2-RBF, $k(\mathbf{x}, \mathbf{y}) = e^{-\frac{d^2(\mathbf{x},\mathbf{y})}{2\sigma^2}}$ avec $d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i \frac{(x_i - y_i)^2}{(x_i + y_i)}}$

$\Rightarrow$ Besoin d'un schéma de recherche rapide pour la distance χ^2

Mise en évidence de l'intérêt de la distance χ^2

4 500 images de la base COREL : sélection de 2 dimensions de l'espace BoW couleur et texture

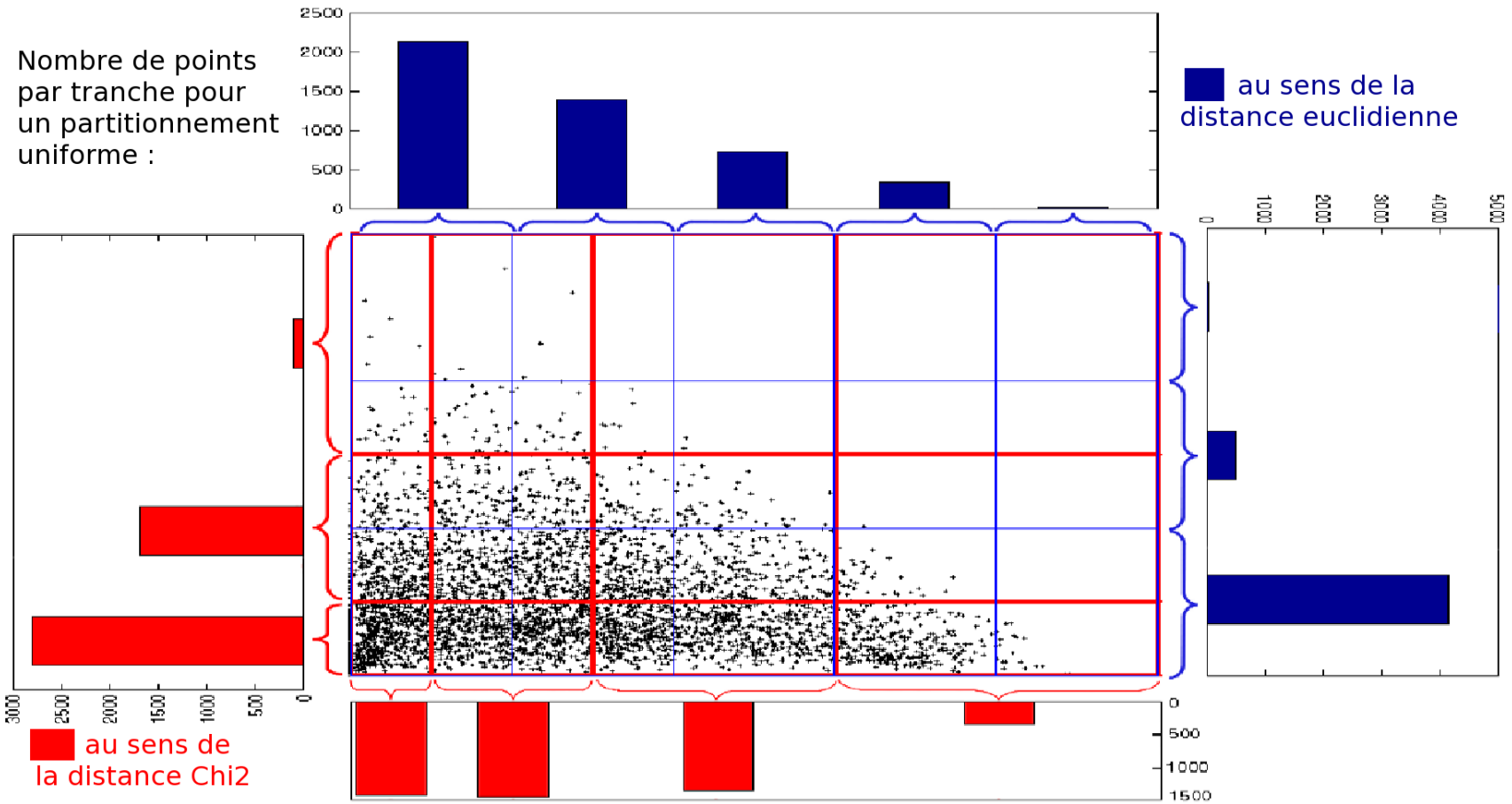
Nombre de points par tranche pour un partitionnement uniforme :



Découpage avec un pavage uniforme
$\Rightarrow$ mauvaise répartition des cellules

Mise en évidence de l'intérêt de la distance χ^2

4 500 images de la base COREL : sélection de 2 dimensions de l'espace BoW couleur et texture



Découpage avec un pavage non uniforme
$\Rightarrow$ meilleure repartition des cellules

CHI2-LSH : LSH pour la distance χ^2

Objectif

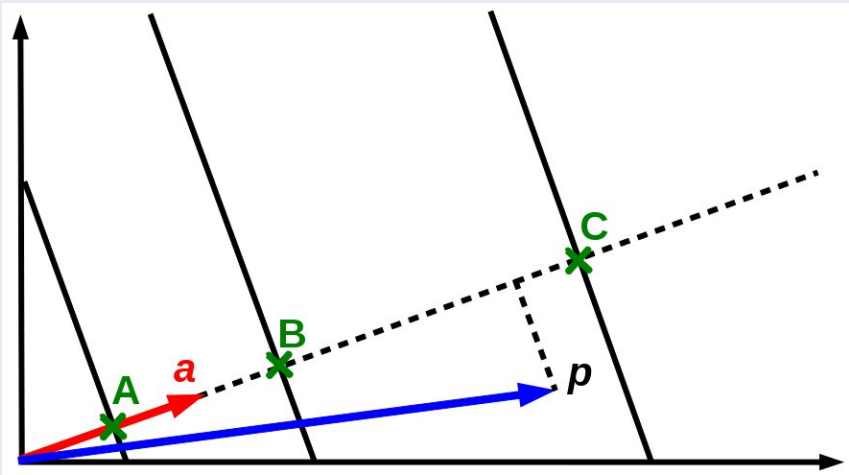
Recherche rapide selon la distance χ^2 et approximation du Noyau χ^2-RBF.

Principe

- Projection des données sur **a** : une direction aléatoire de l'espace
- Découpage régulier de l'espace sur l'axe projeté (au sens du χ^2)

$$\chi^2(A, B) = \chi^2(B, C)$$

- Utilisation arbre de recherche, ...



CHI2-LSH : LSH pour la distance χ^2

Objectif

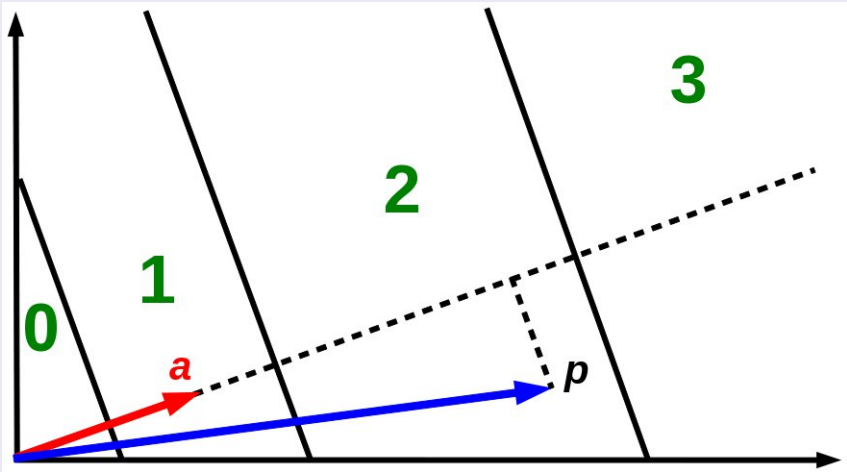
Recherche rapide selon la distance χ^2 et approximation du Noyau χ^2-RBF.

Principe

- Projection des données sur **a** : une direction aléatoire de l'espace
- Découpage régulier de l'espace sur l'axe projeté (au sens du χ^2)

$$\chi^2(A, B) = \chi^2(B, C)$$

- Utilisation arbre de recherche, ...



Fonction de hachage

$$h_{\mathbf{a},b}(\mathbf{p}) = \lfloor \frac{\sqrt{\frac{8\mathbf{a}.\mathbf{p}}{W^2} + 1} - 1}{2} + b \rfloor$$

CHI2-LSH

Système de hachage complet

Pour augmenter le partitionnement des données :

- concaténation de M fonctions de hachage

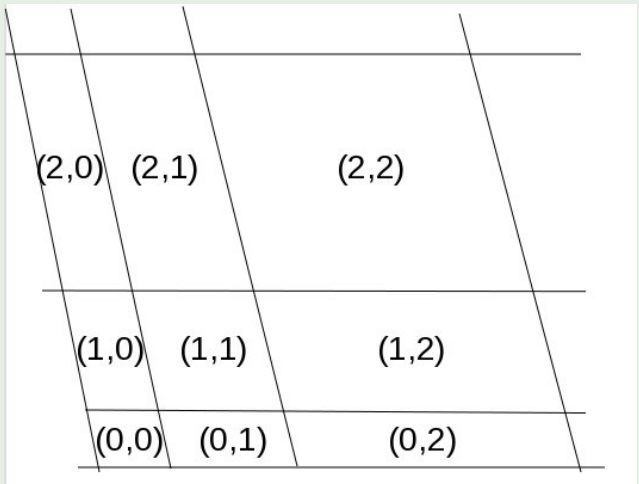
Pour réduire les effets de bords :

- L tables de hachage

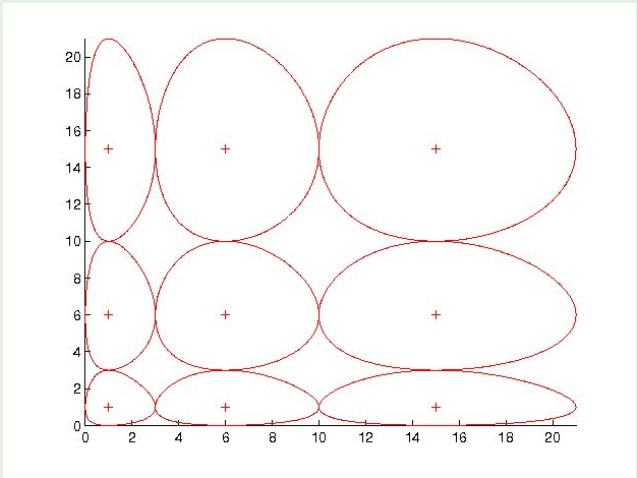
Pour réduire l'occupation mémoire :

- Extension au Multi-Probe LSH : visite des T meilleurs Bucket (probes) [lv2007]

Réseau CHI2-LSH



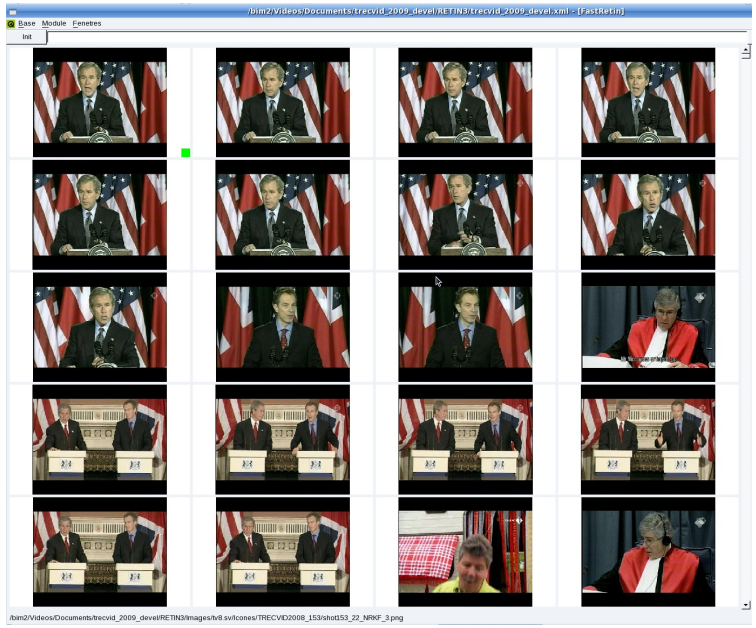
Réseau sphérique χ^2



Evaluation 1 : précision VS temps de recherche

- Bases :
  - 158 000 KeyFrames de TrecVid 2007 à 2009
- Index : BoW 128 dimensions
  - 64 couleurs Lab
  - 64 textures (filtres de Gabor)
- Protocole de test :
  - Verité terrain : recherche exacte 20-NN
  - Précision recherche rapide 20-NN

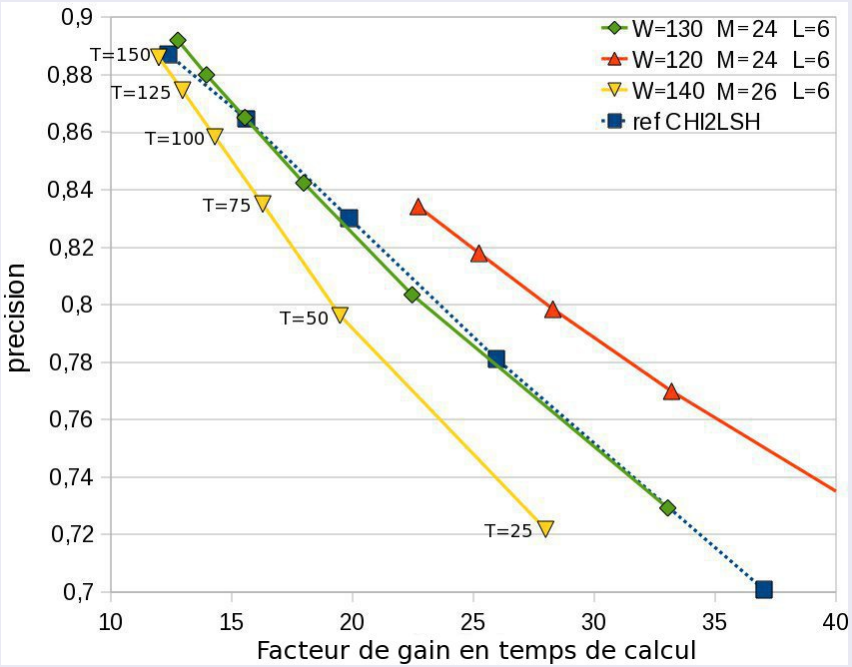
Exemple de recherche linéaire (240 ms)

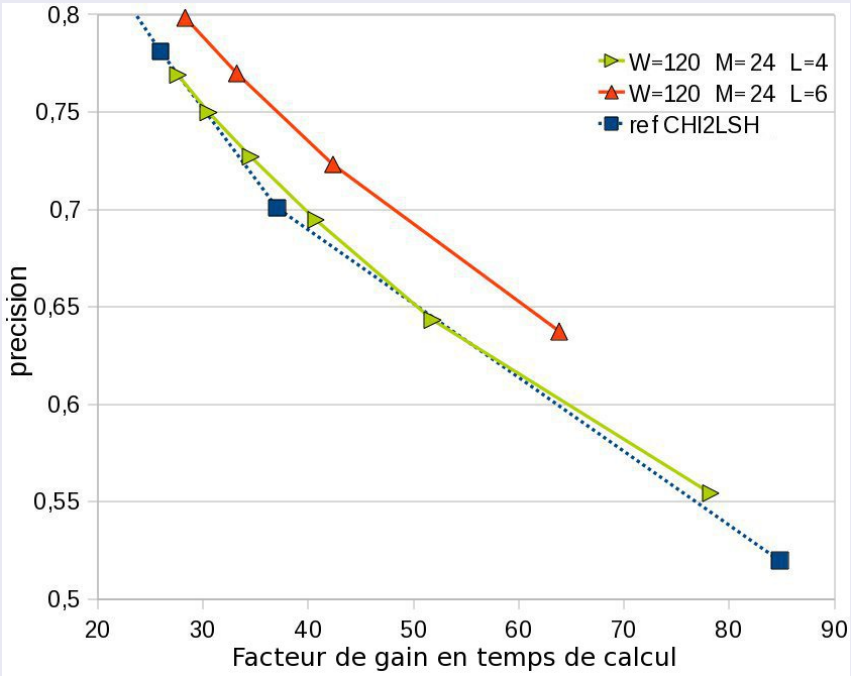


Exemple de recherche rapide (8 ms)

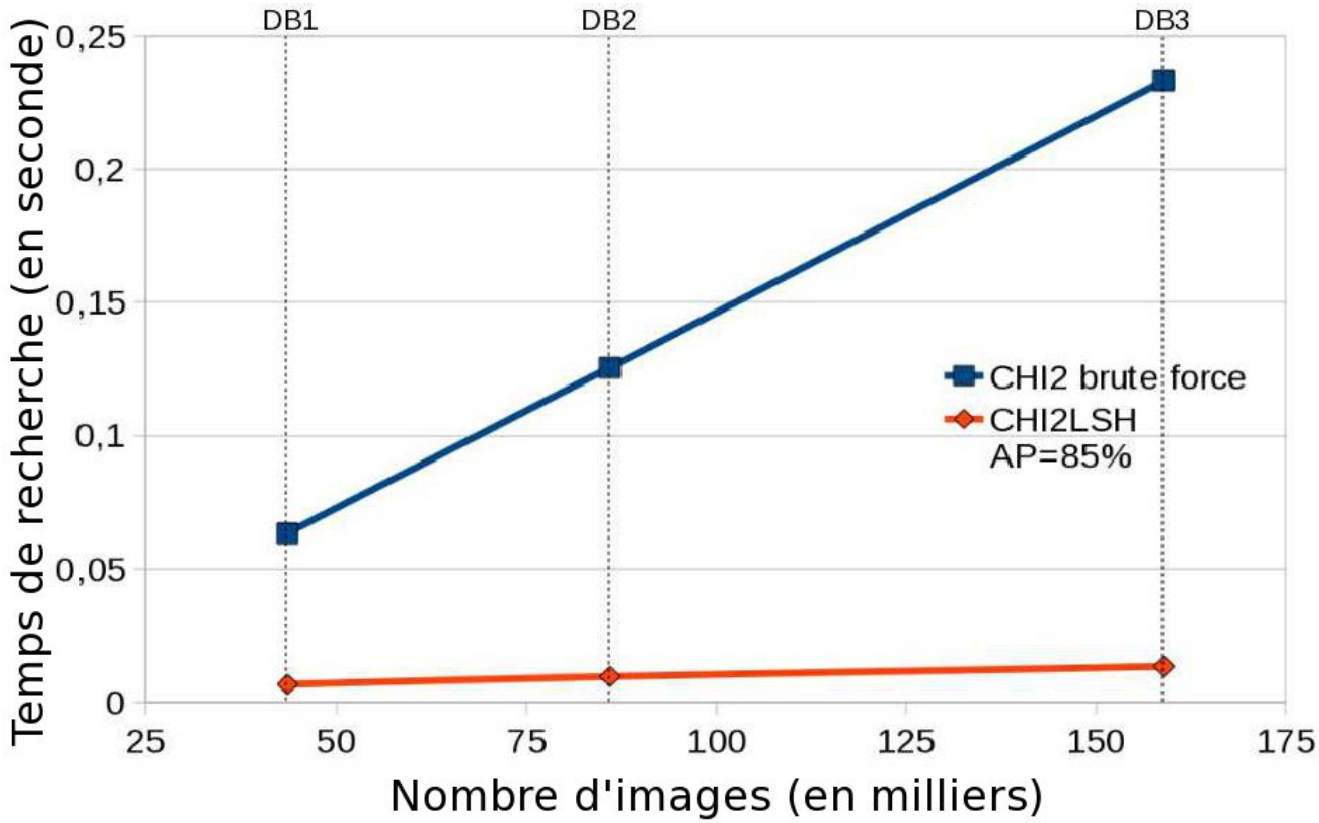
Evaluation 1 : précision VS temps de recherche

Paramétrage du système : W, M, L, T





Evaluation 1 : évaluation de la complexité



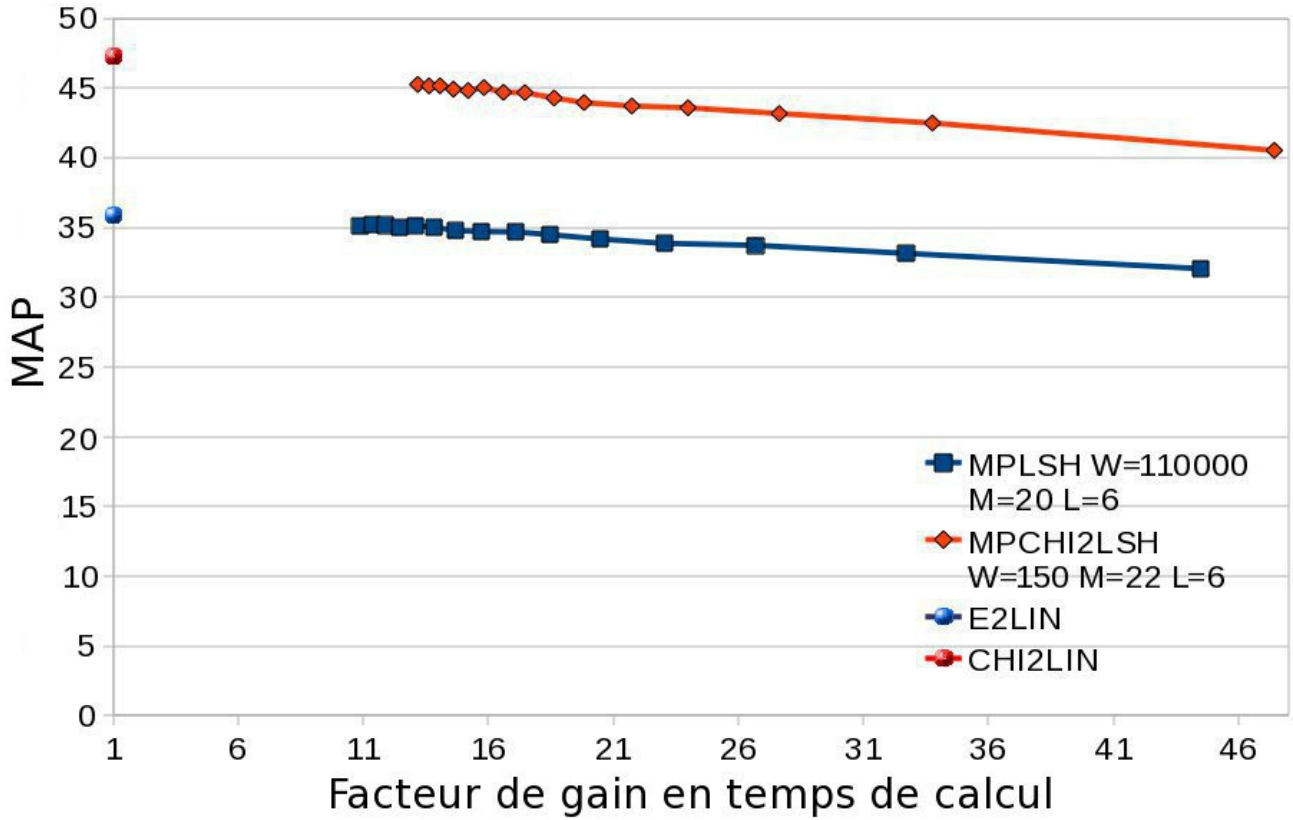
Evolution des temps de recherche en fonction de la taille de la base

Base	Taille	Exacte	CHI2-LSH
DB1 : TrecVid 2007	43 000	63 ms	6,67 ms
DB2 : DB1 + TrecVid 2008	86 000	126 ms	9,68 ms
DB3 : DB2 + TrecVid 2009	158 000	233 ms	13,44 ms

Evaluation 2 : comparaison E2-LSH & CHI2-LSH

- Base issue de COREL :
  - 154 catégories
  - 99 images par catégories
  - $\Rightarrow$ 15 246 images
- Index : BoW 128 dimensions
  - 64 couleurs Lab
  - 64 textures (filtres de Gabor)
- Protocole de test : Leave One out
  - classifieur K-NN ($K = 20$)
  - score : precision moyenne
- Méthodes comparées :
  - E2-Lin
  - E2-LSH
  - CHI2-Lin
  - CHI2-LSH

Evaluation 2



Bilan

- distance χ^2 mieux adaptée que l_2 : Gain de 30% (MAP 47 $\Leftrightarrow$ 36)
- CHI2-LSH et E2-LSH performance similaire (précision / gain en temps de recherche)

Bilan

Contributions

- Nouvelle fonction de hachage adaptée à la représentation BoW
- Optimisation Multi-Probe pour réduire l'occupation mémoire :

$\Rightarrow$ Max (30 Go RAM) : 50 millions d'images (dim 128), 200 M (dim 32)

- Propriétés « Local Sensitivity » démontrées pour notre χ^2-LSH :

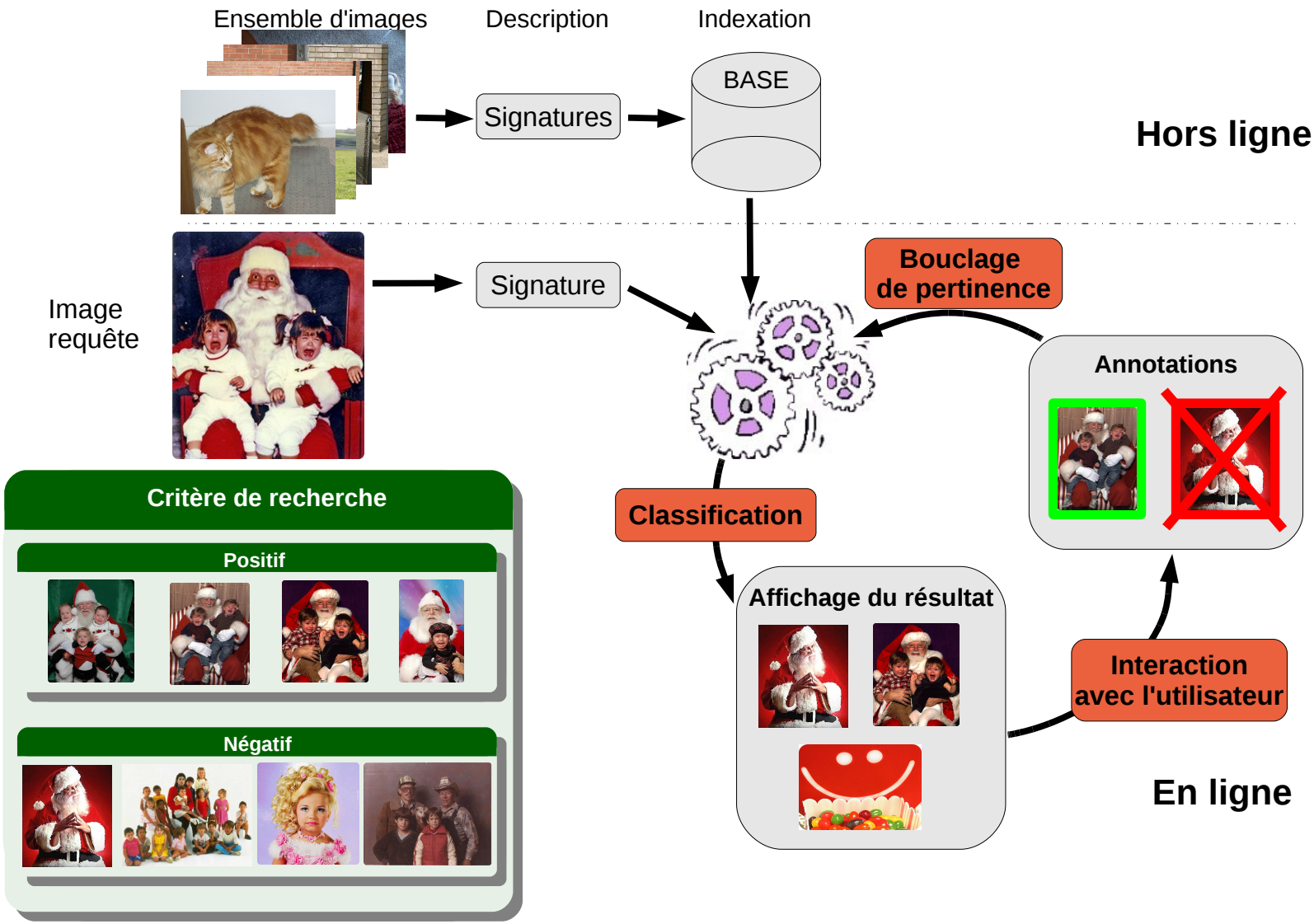
Si $\chi^2(p,q) \leq R$ alors $Pr_{\mathcal{H}}\left[\left\lfloor \frac{\sqrt{\frac{8\mathbf{a}.\mathbf{p}}{w^2}+1}-1}{2} + b \right\rfloor = \left\lfloor \frac{\sqrt{\frac{8\mathbf{a}.\mathbf{q}}{w^2}+1}-1}{2} + b \right\rfloor\right] \geq p_1$

Si $\chi^2(p,q) \geq cR$ alors $Pr_{\mathcal{H}}\left[\left\lfloor \frac{\sqrt{\frac{8\mathbf{a}.\mathbf{p}}{w^2}+1}-1}{2} + b \right\rfloor = \left\lfloor \frac{\sqrt{\frac{8\mathbf{a}.\mathbf{q}}{w^2}+1}-1}{2} + b \right\rfloor\right] \leq p_2$

Avec $c = 1+\epsilon$ et $p_1 > p_2$.

Plan

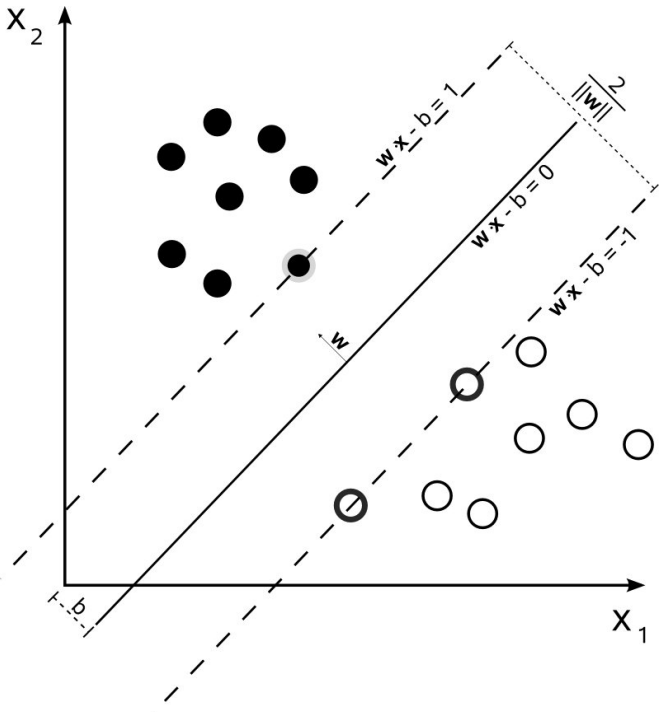
Moteur de recherche interactif [TC01]



Technique d'apprentissage : Classifieur SVM [VV98]

Ensemble d'apprentissage & Données à classer

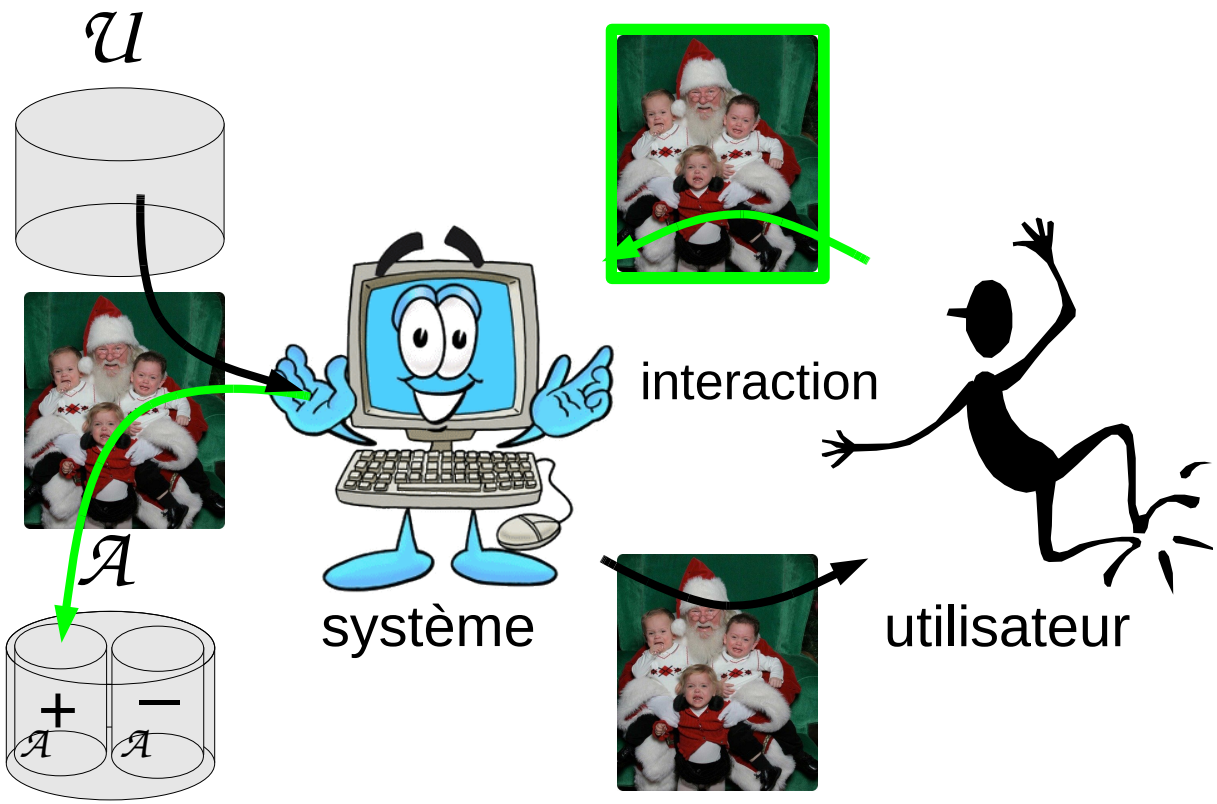
- $\mathcal{A} = \{(\mathbf{x}_i, y_i)_{i=1,N} \mid y_i \neq 0\}$
  - $y_i = +1$ exemple positif.
  - $y_i = -1$ exemple négatif.
- $\mathcal{U} = \{(\mathbf{x}_i, y_i)_{i=1,N} \mid y_i = 0\}$



SVM :

- Minimiser $\frac{||\mathbf{w}||^2}{2}$ sous la contrainte $(\forall \mathbf{x_i} \in \mathcal{A})\ y_i(\langle \mathbf{w}, \phi(\mathbf{x_i}) \rangle + b) \geq 1$
- Fonction de pertinence $f_{\mathcal{A}}(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b$
- Estimer la pertinence de $\mathcal{U}$

Construction ensemble apprentissage :



Apprentissage intéractif : Construction iterative base d'apprentissage.

- Système sélection une image $\mathbf{x}_s \in \mathcal{U} \Rightarrow$ présente $\mathbf{x}_s$ à l'utilisateur.
- Utilisateur annote l'image : si $\mathbf{x}_s$ est pertinente : $y_s = +1$ sinon $y_s = -1$.
- le couple $(\mathbf{x}_s, y_s) \cup \mathcal{A}$.
- Construction de $\mathcal{A}$ itérativement en utilisant des intéractions Homme-Machine.

Sélection des images à annoter : apprentissage actif

But

Quelles images annoter pour améliorer au mieux la recherche ?

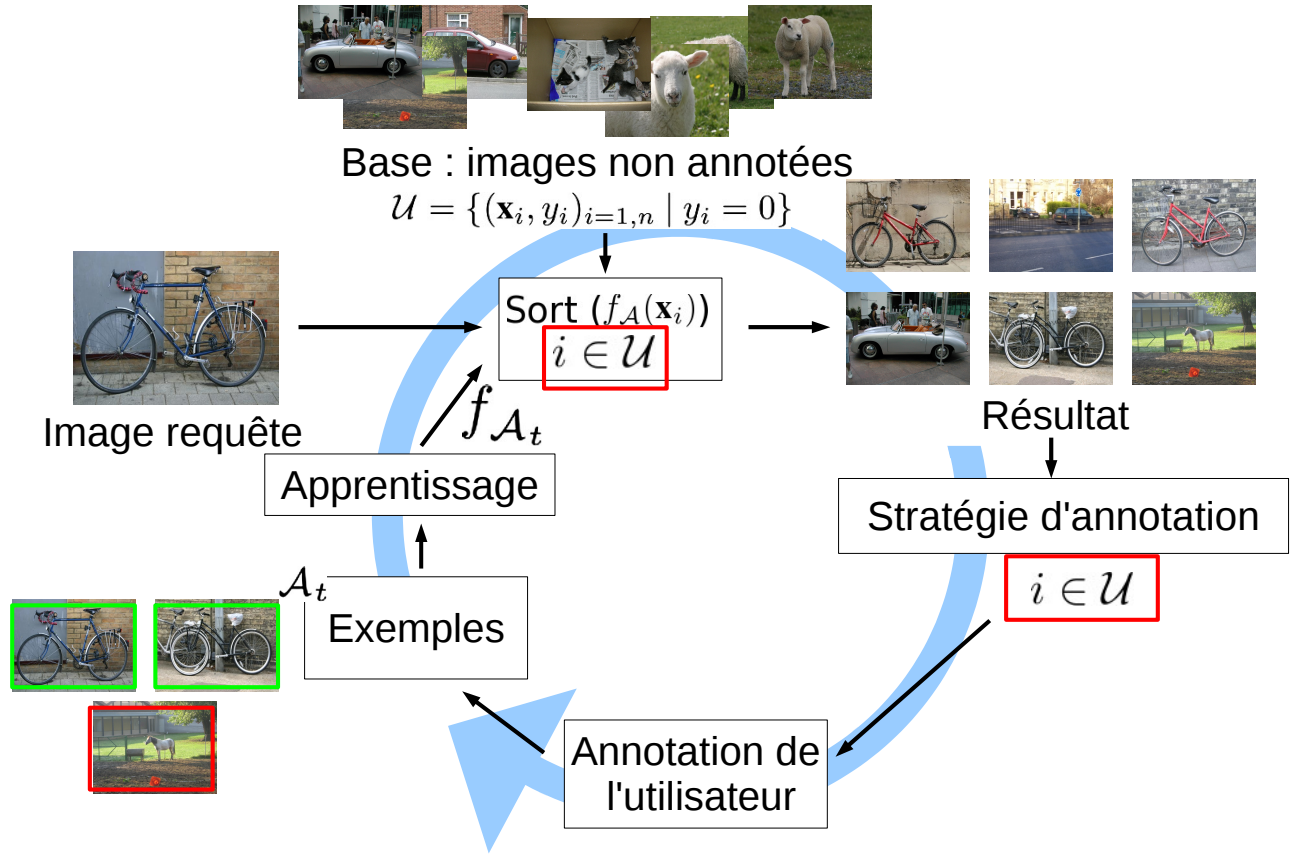
Lesquelles choisir ?

- Les plus incertaines [Tong02] et les plus diversifiées : Diversité des Angles [Brinker03] :

$$i^\star = \arg\min_{\mathbf{x}_i \in \mathcal{U}} (\lambda |f_{\mathcal{A}}(\mathbf{x}_i)| + (1-\lambda) \max_{\mathbf{x}_j \in \mathcal{A}} \frac{k(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{k(\mathbf{x}_i, \mathbf{x}_i) k(\mathbf{x}_j, \mathbf{x}_j)}})$$

- Autres stratégies : les plus pertinentes, ... [Zhou03]

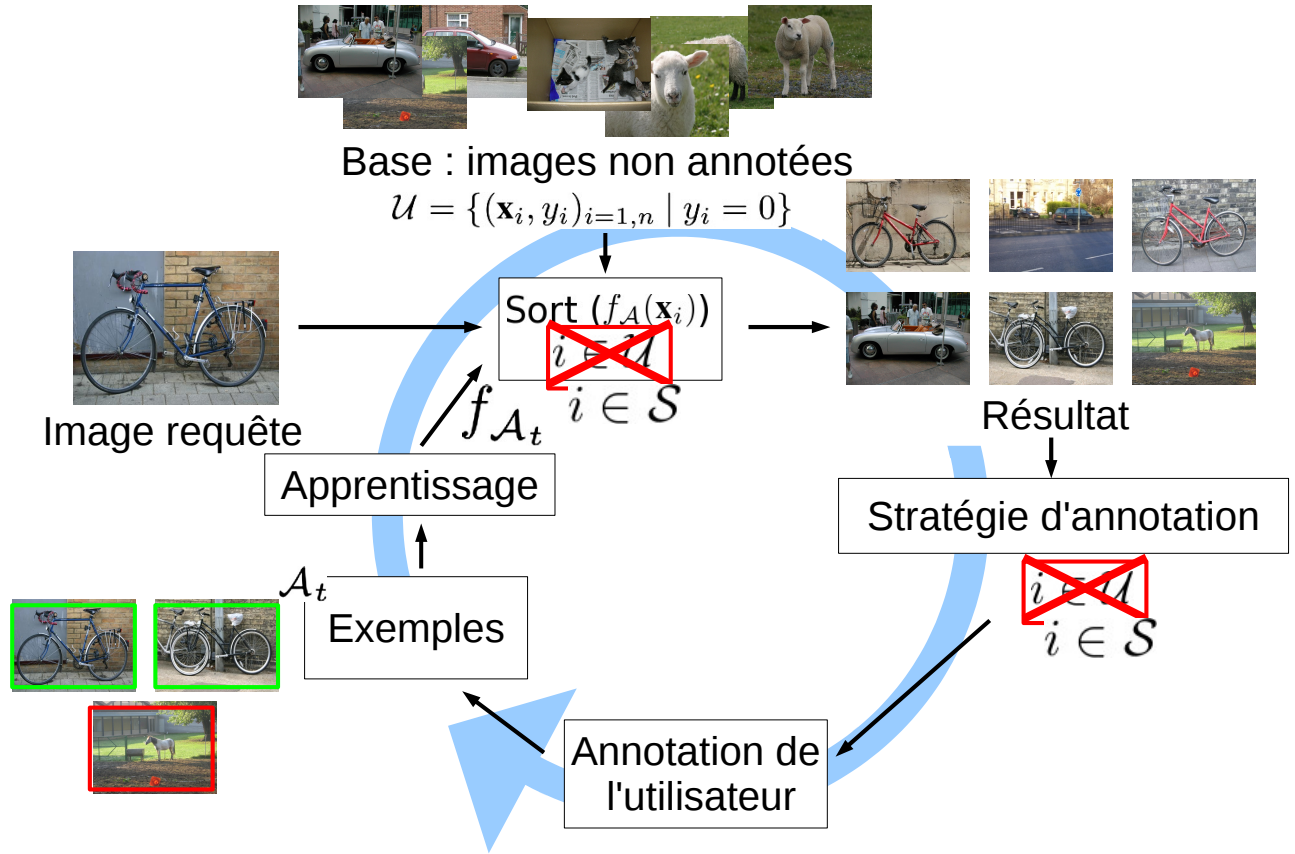
Identification du problème de complexité



- Complexite linéaire par rapport à la taille de la base :

  $\Rightarrow$ inadapté aux grandes bases d'images

Identification du problème de complexité



Idée : sous échantillonnage

- Considérer uniquemnet un sous ensemble $\mathcal{S}$ au lieu de $\mathcal{U}$

  avec $N < |\mathcal{S}| << n = \mathcal{U}$
- Comment construire $\mathcal{S}$?

Comment accélérer la recherche ?

Méthodes

- Sous échantillonnage de $\mathcal{U}$ – sélection aléatoire ?
- Méthode de clustering : Hierarchical sampling [Panda06]
- Utilisation d'un M-Tree [Crucianu07]

Difficulté

- fonction de pertinence du SVM complexe :

$$f_{\mathcal{A}}(\mathbf{x}) = \sum_{i=0}^{|\mathcal{A}|} y_i \alpha_i K(\mathbf{a}_i, \mathbf{x})$$

Notre stratégie : SALSAS

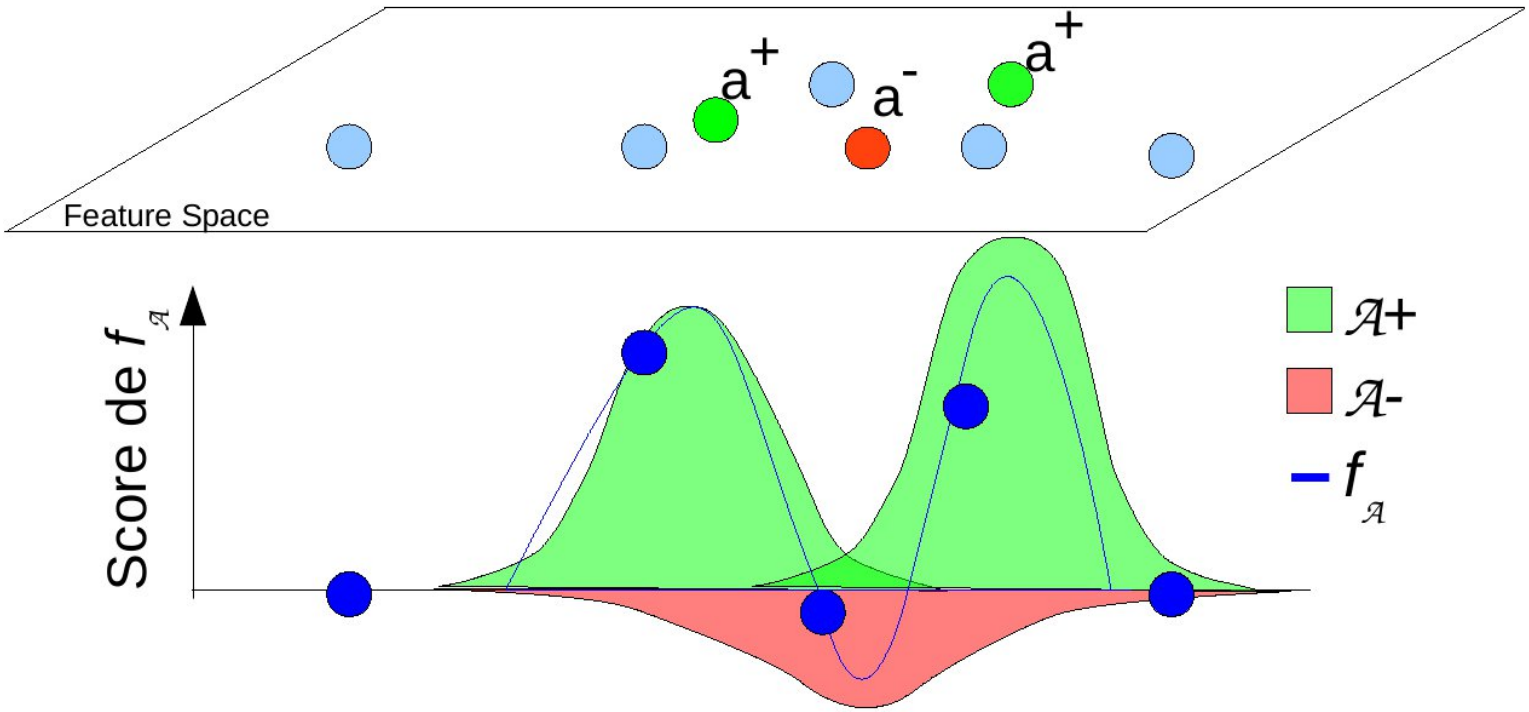
- échantillonnage basée sur une recherche KNN optimisée (LSH)
- approximation de $f_{\mathcal{A}}$

Stratégie d'échantillonnage

Décomposition de $f_{\mathcal{A}}$

$$f_{\mathcal{A}}(\mathbf{x}) = \sum_{p \in \mathcal{A}^+} \alpha_p K(\mathbf{a}_p^+, \mathbf{x}) - \sum_{n \in \mathcal{A}^-} \alpha_n K(\mathbf{a}_n^-, \mathbf{x}) = f_{\mathcal{A}^+}(\mathbf{x}) - f_{\mathcal{A}^-}(\mathbf{x})$$

avec $\mathcal{A}^+$: exemples positifs $\mathcal{A}^-$: exemples négatifs

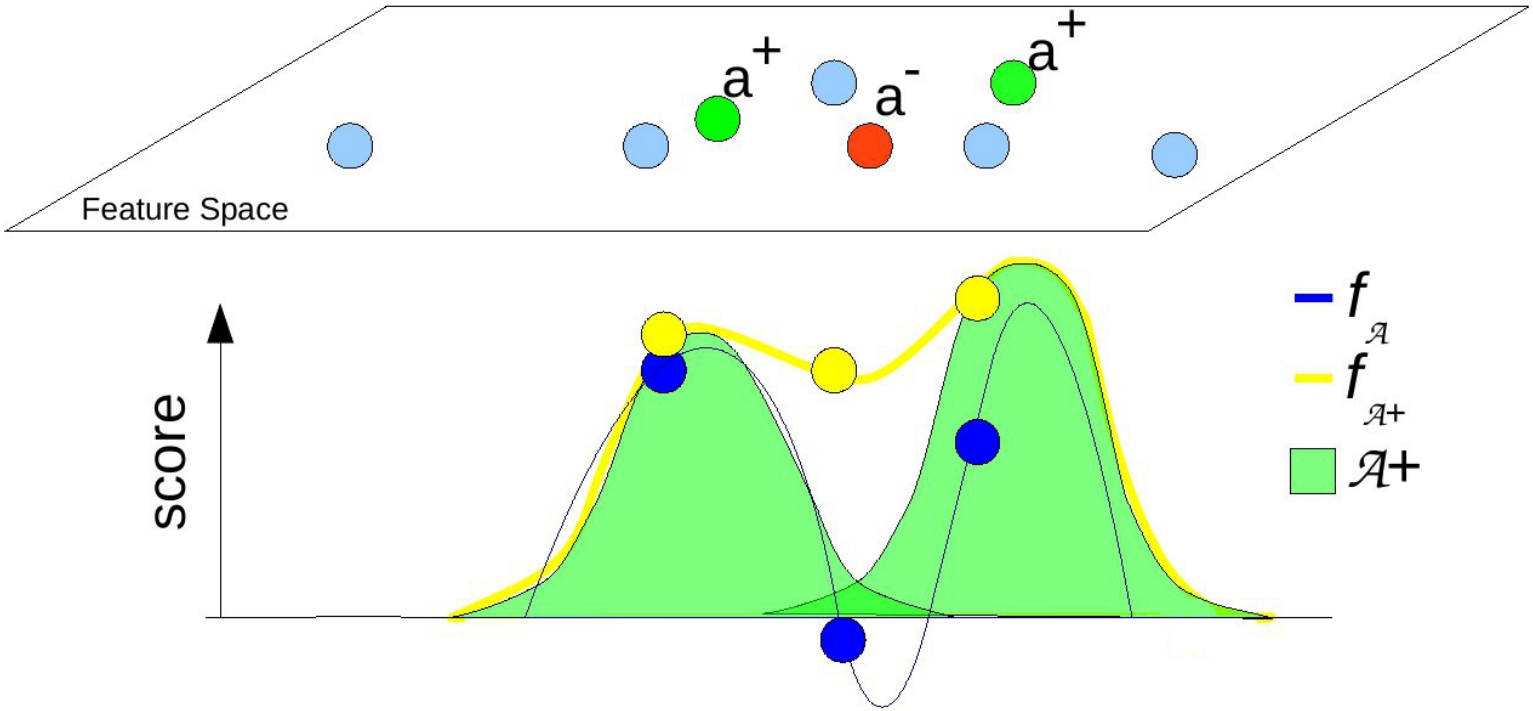


Stratégie d'échantillonnage

Recherche TopN : approximation de la maximisation de $f_{\mathcal{A}}$

- score $f_{\mathcal{A}}(x)$ élevé $\Rightarrow$ $f_{\mathcal{A}^+}(x)$ élevé

$\Rightarrow$ Maximiser $f_{\mathcal{A}}$ avec Maximisation de $f_{\mathcal{A}^+}$

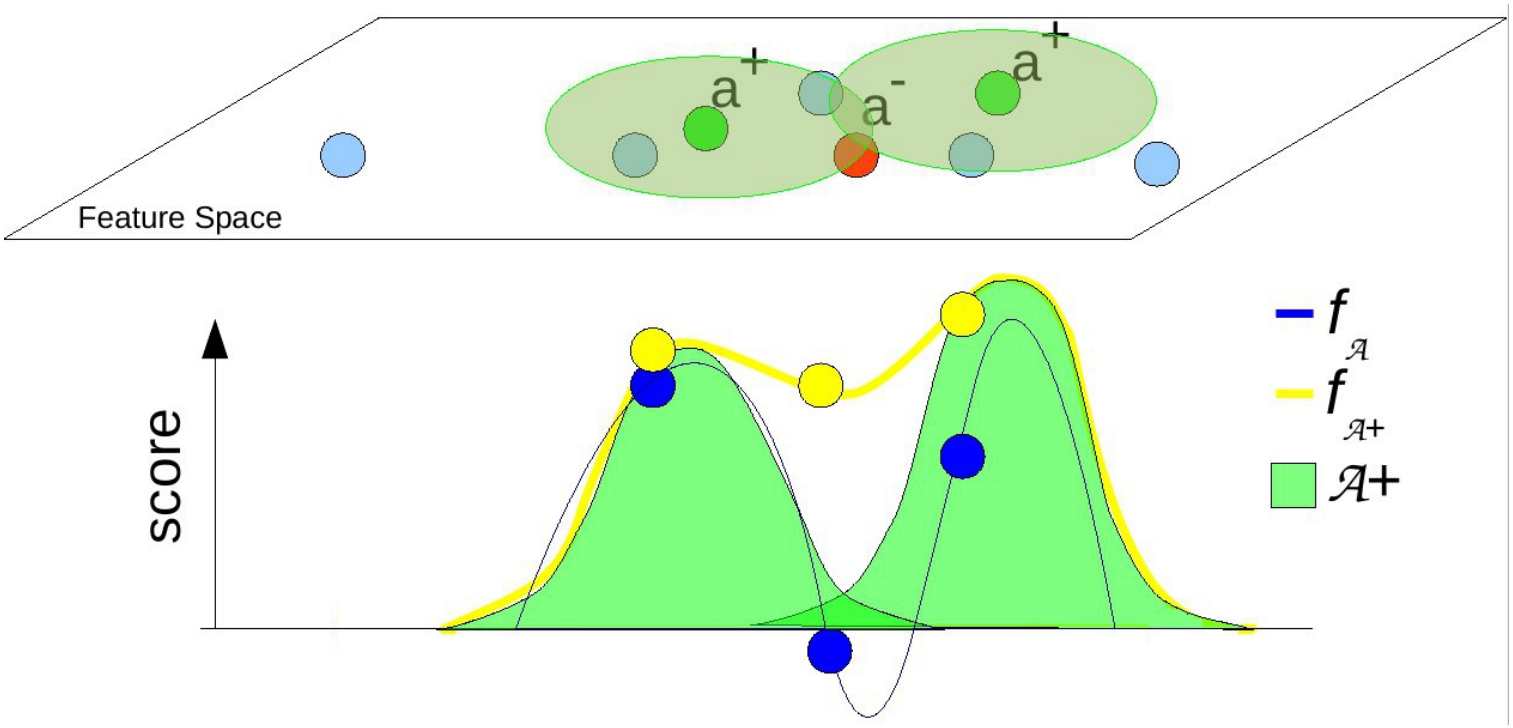


Stratégie d'échantillonnage

SALSAS : construction du pool $\mathcal{S}$

- Maximisation rapide de $f_{\mathcal{A}^+}$

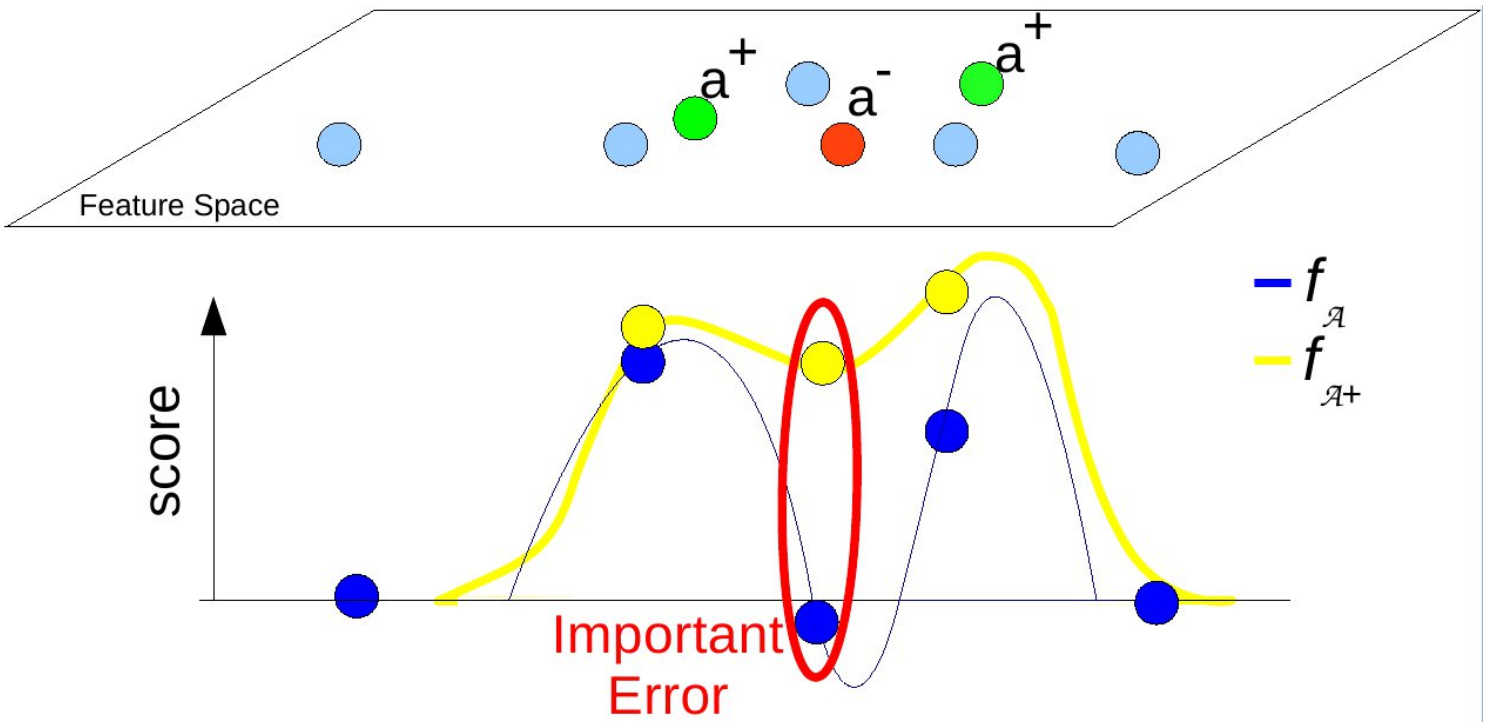
$\Rightarrow \mathcal{S} = \{\mathbf{x} \in \mathcal{U} \mid \mathbf{x} = KNN(\mathbf{a})\}$ avec $\mathbf{a} \in \mathcal{A}^+$



Stratégie d'échantillonnage

Elagage

- $f_{\mathcal{A}^+}(x)$ élevé $\not\Rightarrow$ $f_{\mathcal{A}}(x)$ élevé
- Calcul de $f_{\mathcal{A}}$ possible sur $\mathcal{S}$



Stratégie d'annotation

Recherche dans $\mathcal{S}$

- $\mathcal{S} > N \Rightarrow \mathcal{S}$ contient des images incertaines

$\Rightarrow$ Rechercher des images les plus incertaines et les plus diversifiées de $\mathcal{S}$:

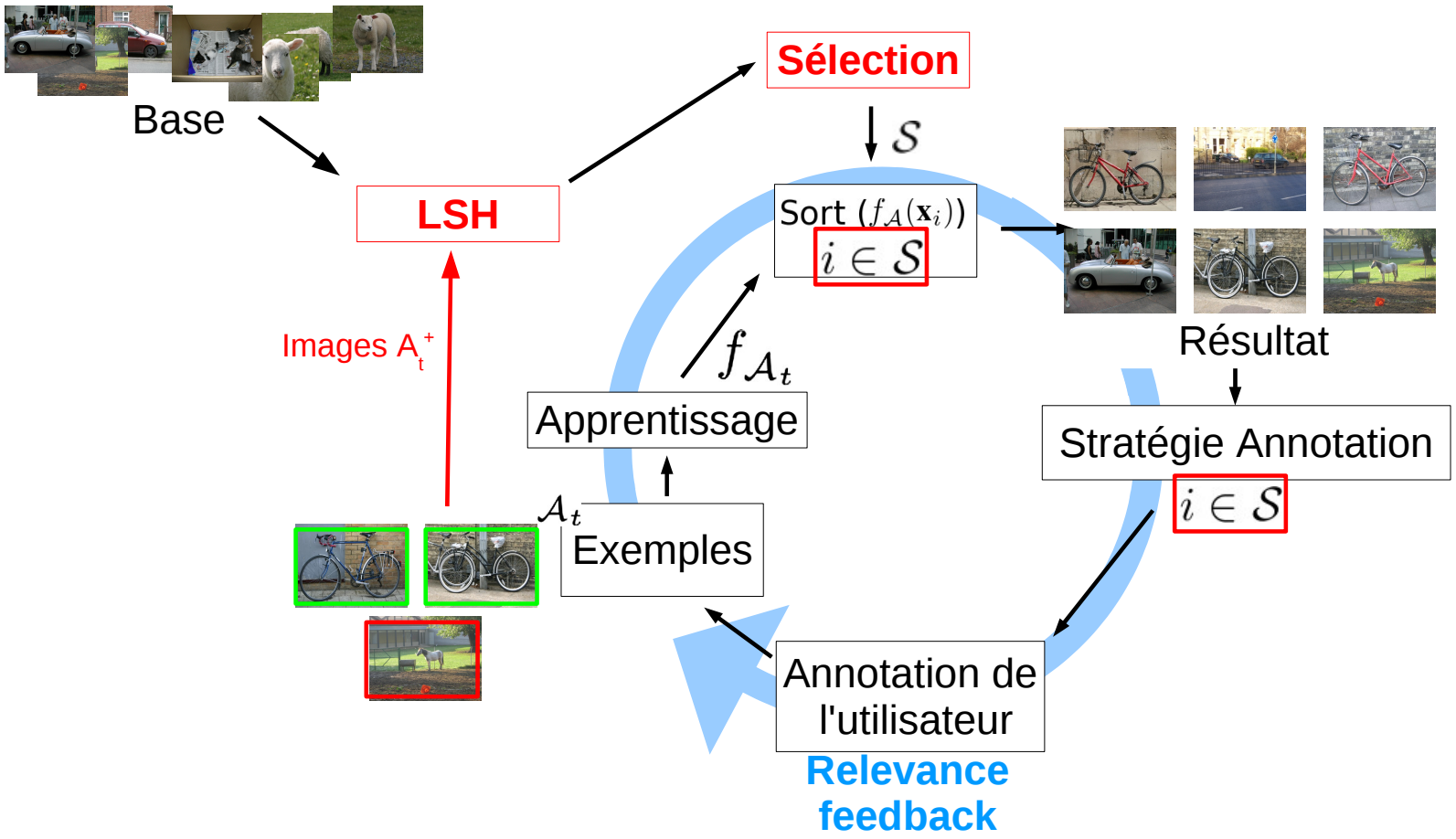
$$i^\star = \underset{\mathbf{x}_i \in \mathcal{S}}{\arg\min}(\lambda |f_{\mathcal{A}}(\mathbf{x}_i)| + (1-\lambda) \max_{\mathbf{x}_j \in \mathcal{A}} \frac{k(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{k(\mathbf{x}_i, \mathbf{x}_i) k(\mathbf{x}_j, \mathbf{x}_j)}})$$

Intérêts

- Très rapide : $\mathcal{S}$ déjà calculé
- Rééquilibrage du problème :
  - Grandes bases $\Rightarrow$ |images non pertinentes| >> |images pertinentes|

  Recherche dans un voisinage des images pertinentes $\Rightarrow$ plus d'images pertinentes sélectionnées

Notre algorithme : SALSAS

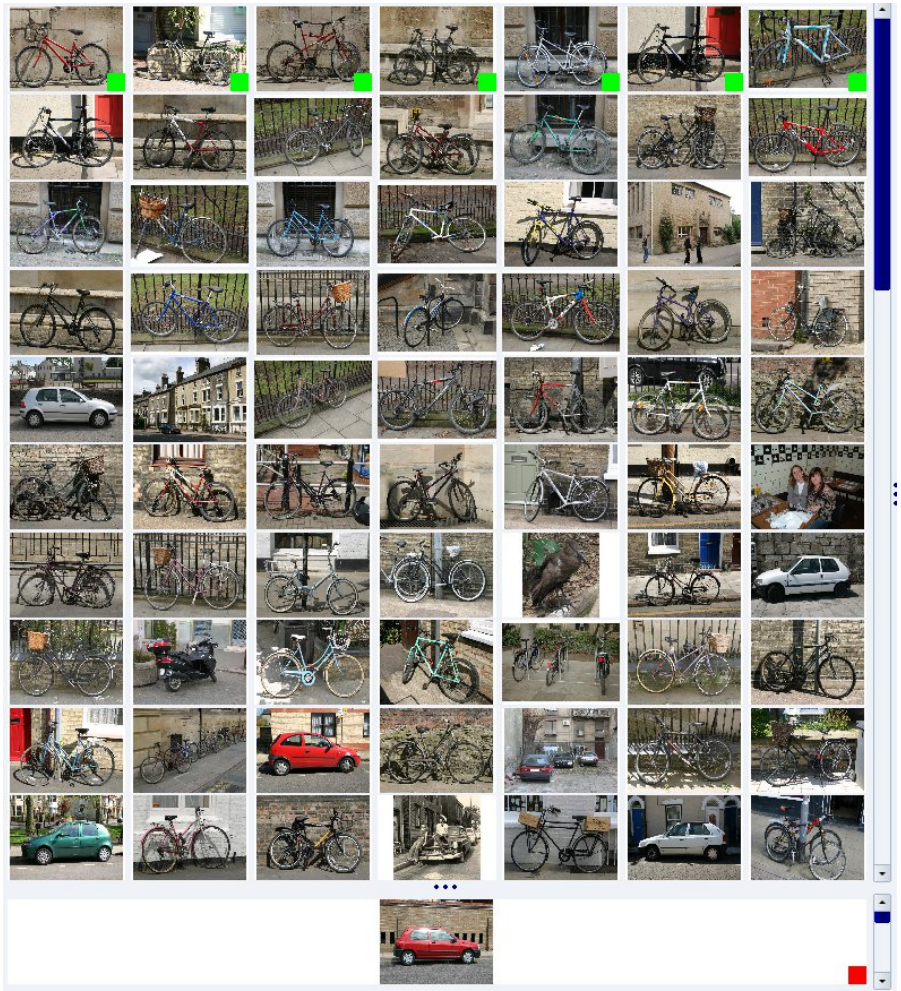


Mise à jour de $\mathcal{S}$:

- Ajout des KNN des nouvelles images annotées positivement
- Pour être efficace : LSH

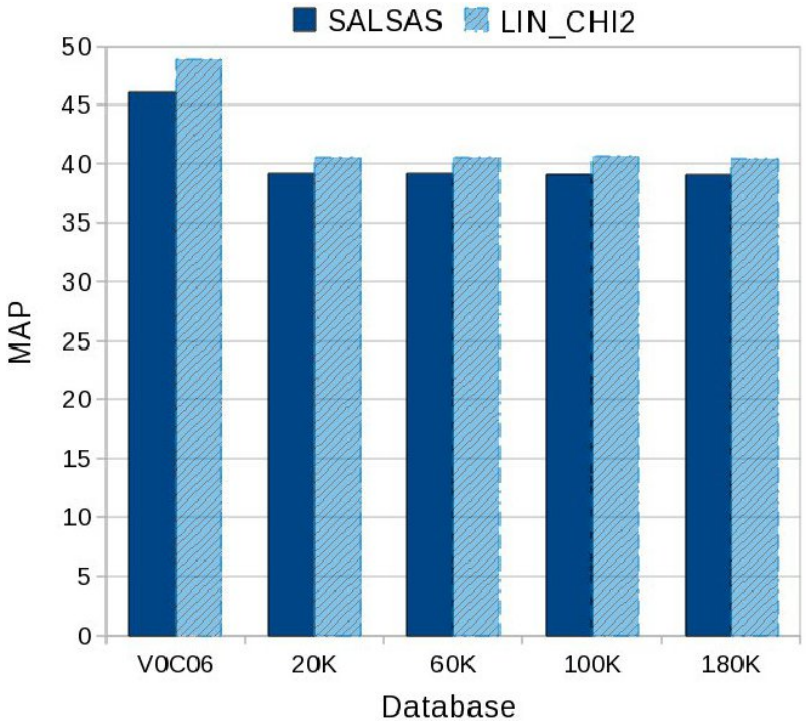
Exemples

- Base : 180 000 images (VOC 2006, 2007, 2008 + TrecVid 2007, 2008, 2009)
- Index : BoW 128 dimensions
  - 64 couleurs Lab
  - 64 textures (filtres de Gabor)
- Paramètres : 1 annotation par itération, TOP200, recherche 100-NN
- Paramètres LSH : $M = 24$ projections, $L = 4$ tables de hachage, $T = 100$ probes

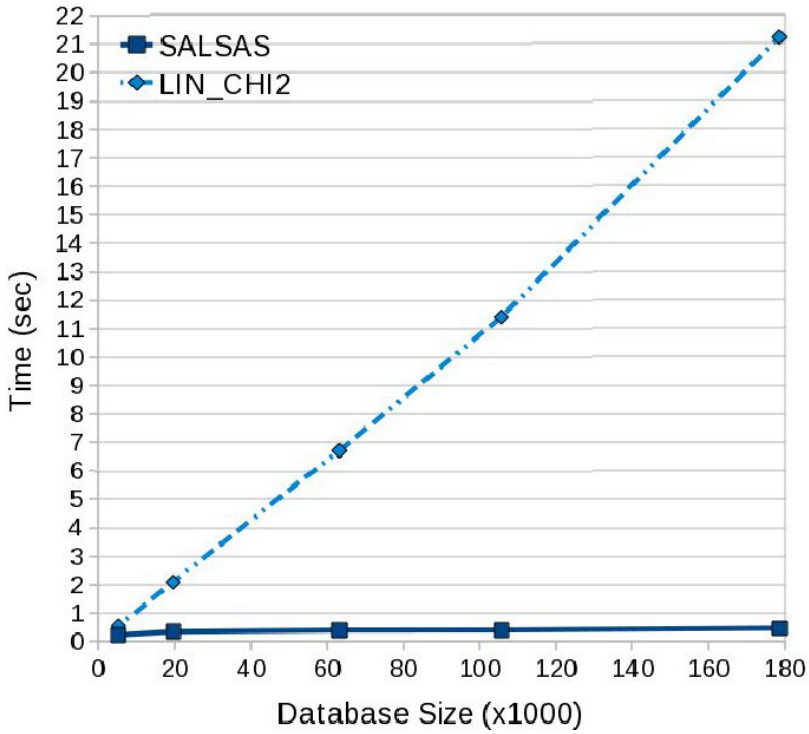


Résultats

- Comparaison de la précision et du temps de recherche entre SALSAS et la recherche exacte
- Session de recherche de 50 itérations
- 5 bases entre 5 000 et 180 000 images
- χ^2-RBF avec $\sigma = 200$
- Paramètres LSH $W = 400$



MAP du TOP200 à la 50^e itération : 10 classes VOC06 pour les 5 bases



Temps de recherche pour 50 itérations vs taille de la base

Bilan

Stratégie pour passage à l'échelle la recherche interactive :

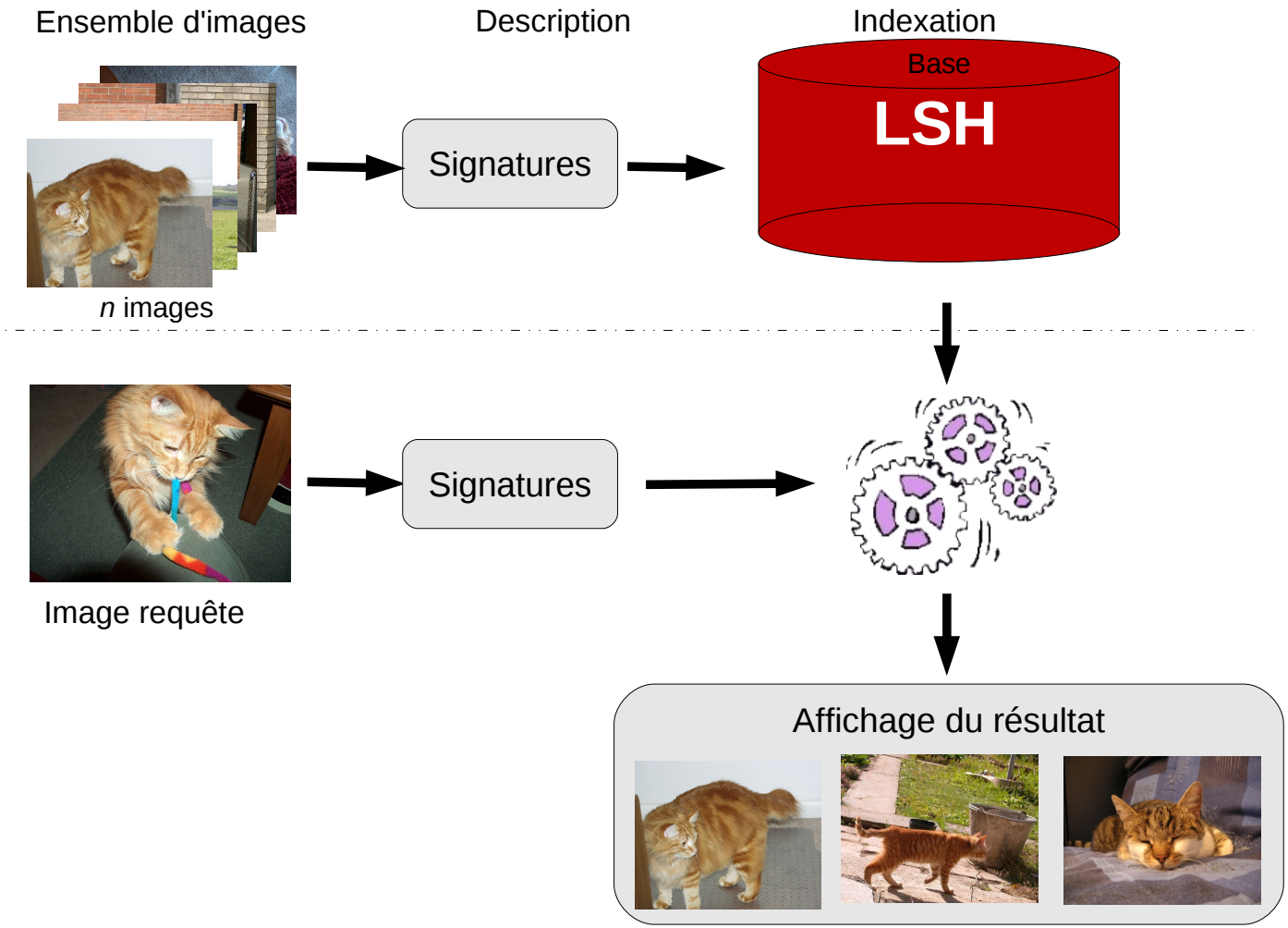
- TopN + annotation : pool d'images $\mathcal{S}$
- Mise à jour du pool rapide (LSH)

$\Rightarrow$ 45 $\times$ plus rapide que la recherche exacte (pour 180K images)

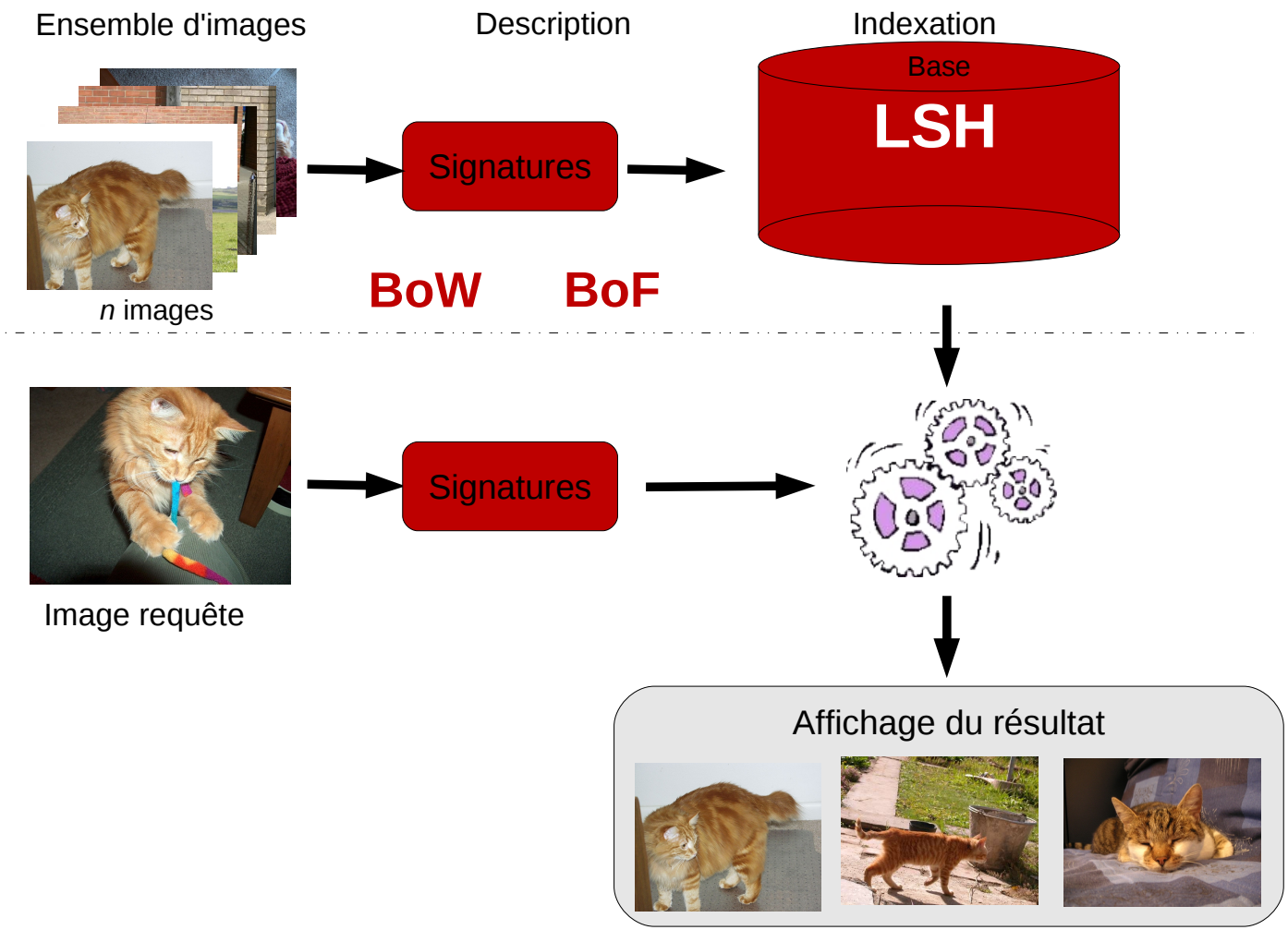
$\Rightarrow$ Qualité de recherche similaire à la recherche exacte

Plan

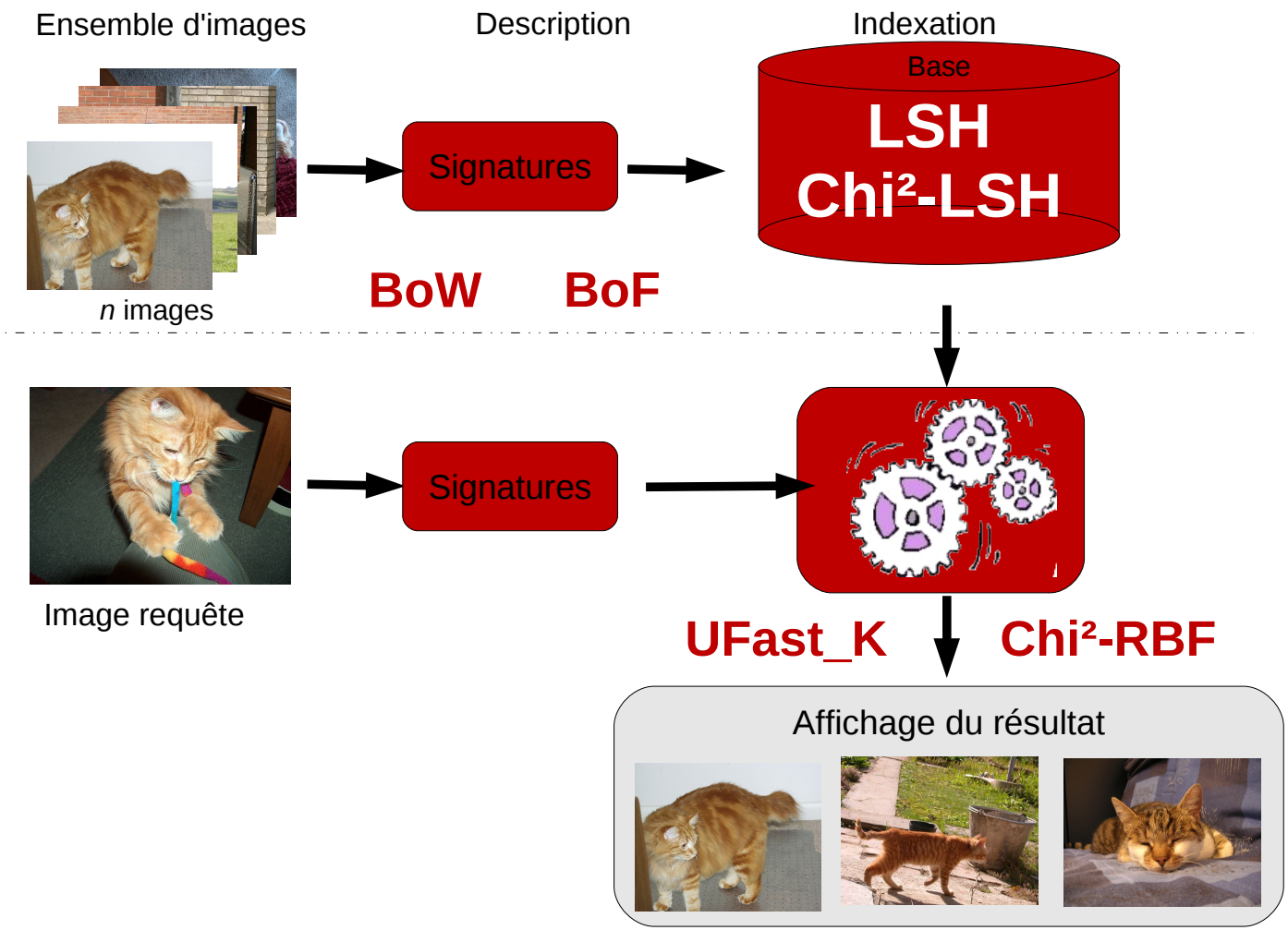
Conclusion


Ensemble d'images
Description
Indexation
Base
LSH
Signatures
n images
Signatures
Image requête
Affichage du résultat

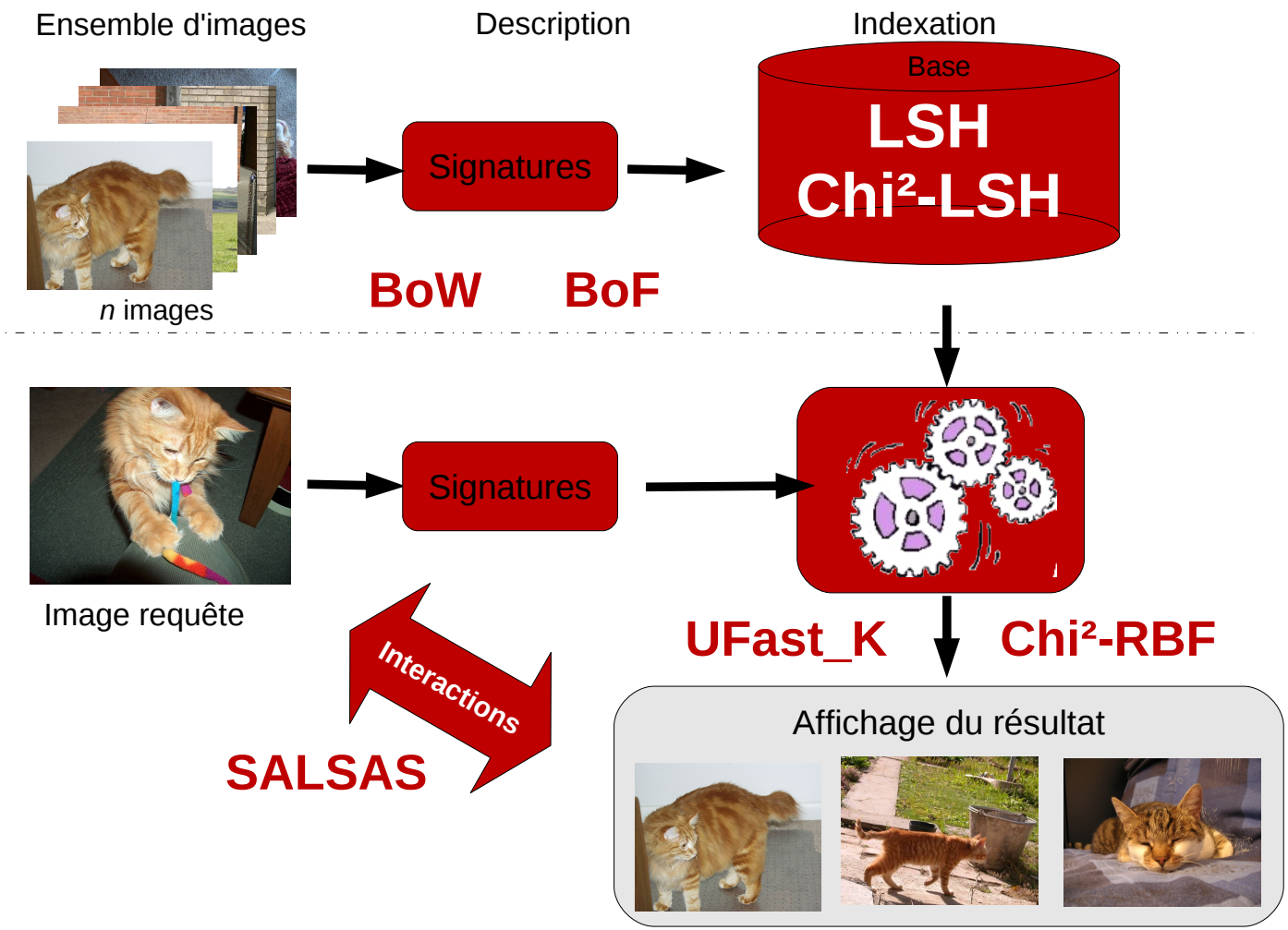
Conclusion


Ensemble d'images
Description
Indexation
Base
LSH
Signatures
BoW
BoF
n images
Signatures
Image requête
Affichage du résultat

Conclusion


Ensemble d'images
Description
Indexation
Base
LSH
Chi²-LSH
Signatures
n images
BoW
BoF
Signatures
Image requête
UFast_K
Chi²-RBF
Affichage du résultat

Conclusion


Ensemble d'images
Description
Indexation
Base
LSH
Chi²-LSH
Signatures
n images
BoW
BoF
Image requête
Signatures
Interactions
SALSAS
UFast_K
Chi²-RBF
Affichage du résultat

Collaborations / Applications

RETIN [Cord, Gosselin, Fournier]

- SALSAS : Journal (Pattern Recognition, *accepted*, 2010) & 2 (ICIP)
- $UFast_K$: (ICPR 08)

consortium IRIM pour les compétitions TrecVid

- Résumés Vidéos (ACM TRECVid Video Summarization Workshop 2008), Classification Vidéos (TREC Video Retrieval Evaluation Online Proceedings 2010)

Eduardo Valle UNICAMP, Campinas, Brésil pour Structures d'index

- Journal : Information Sciences 2010 *in press*, Journal **SOUMIS à** : IEEE transactions on pattern analysis and machine intelligence

Perspectives

Détection de copies proches

- Réduction de l'occupation mémoire des données (ACP, compression)
- Réduction du nombre de PoI pour décrire les images
- Comparaison entre $UFast_K$ et K-LSH [Kulis09]
- Extention de K-LSH au schéma E2-LSH ($|\Phi(X) - \Phi(Y)|^2 = 2 - 2K(X, Y)$) pour noyaux normées

Recherche interactive

- Autres stratégies de sélection
- Amélioration de KDX [Panda05] $\rightarrow$ KDX-LSH
- Utilisation de K-LSH
- Extention de Hashing Hyperplan Queries [Jain10] aux noyaux