

Comprendre les données visuelles à grande échelle

ENSIMAG
2018-2019

KartEEK Alahari & Diane Larlus
December 20th 2018



Organisation du cours

- Séance du jeudi 20 décembre – 8h15 à 11h15
 - Deuxième cours
 - Présentation article 1 + quizz

Cours 2: Représentation d'images avec application à la recherche visuelle - première partie

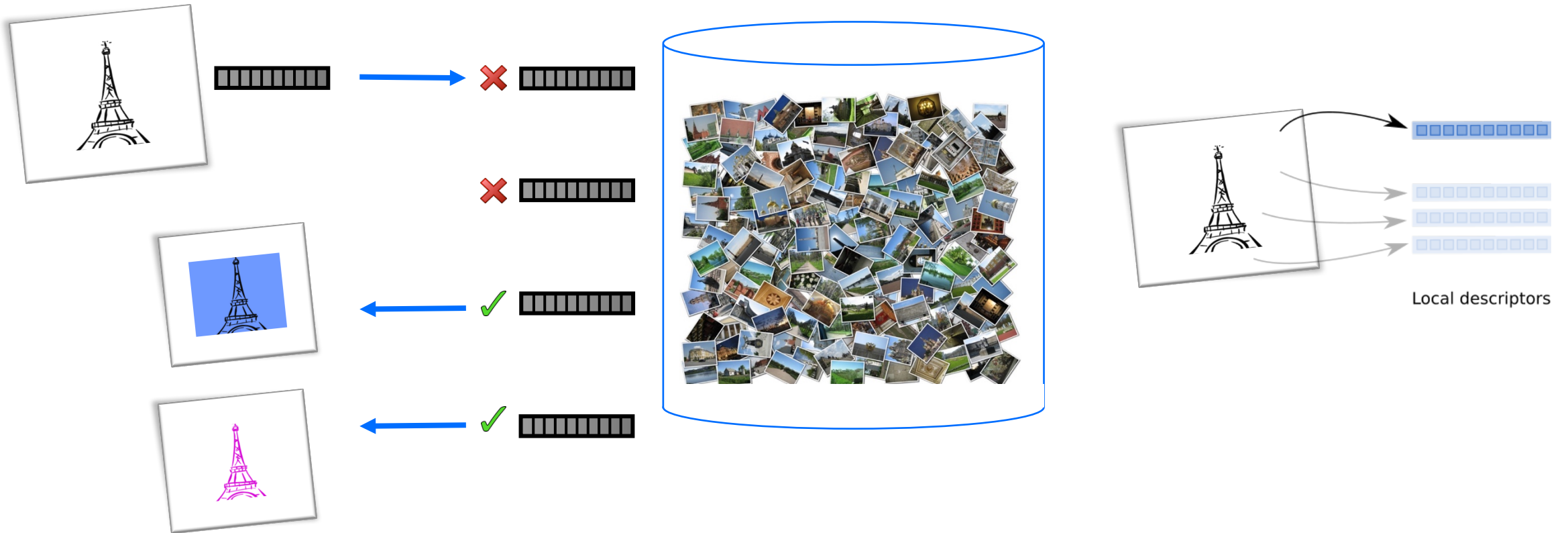
Comprendre les données visuelles à grande échelle
20 décembre 2018

Description locale des images

Comprendre les données visuelles à grande échelle
Cours 2: représentations d'images, 20 décembre 2018

Local representations

It is challenging to capture all these invariances in a global representation



Local representations

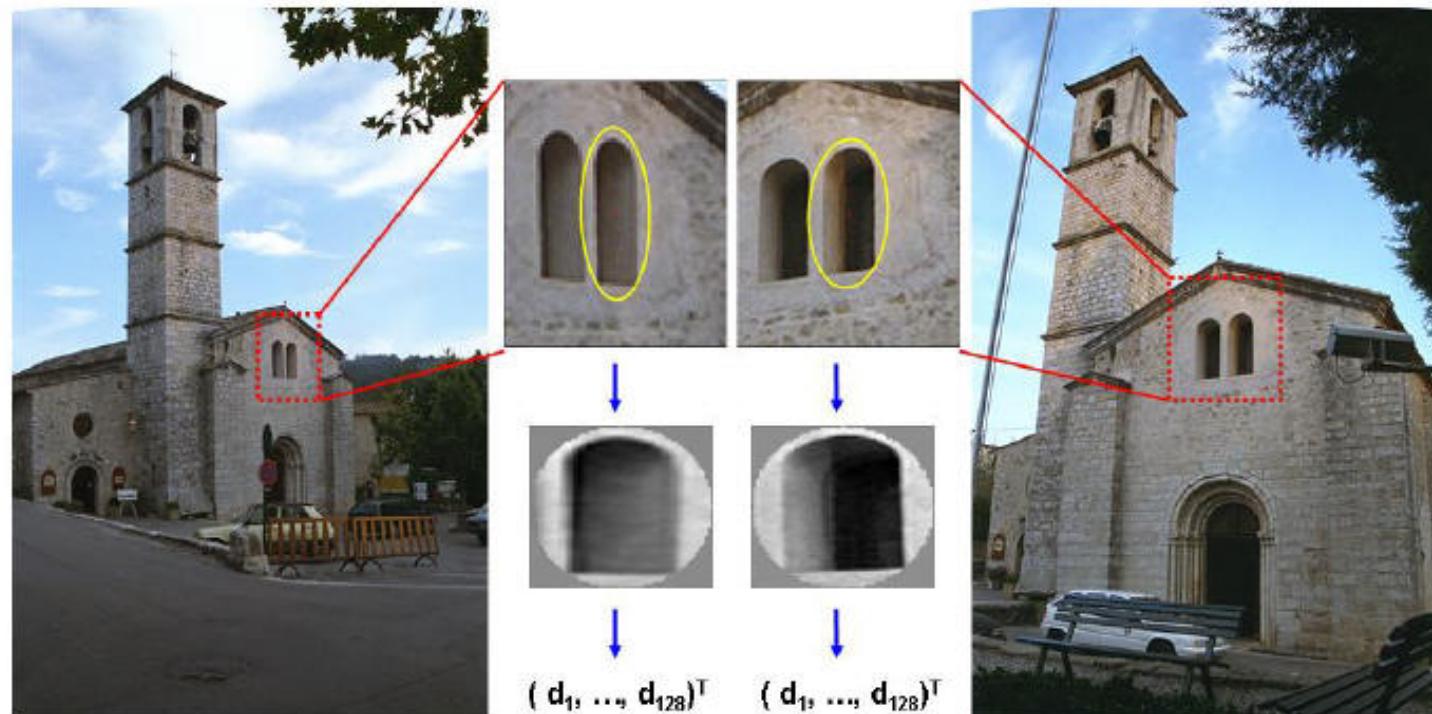
It is challenging to capture all these invariances in a global representation



Local representations seem more adapted but require more sophisticated matching

Local representations

- Local descriptors capture the local appearance
- Carefully chosen locations: **repeatability**



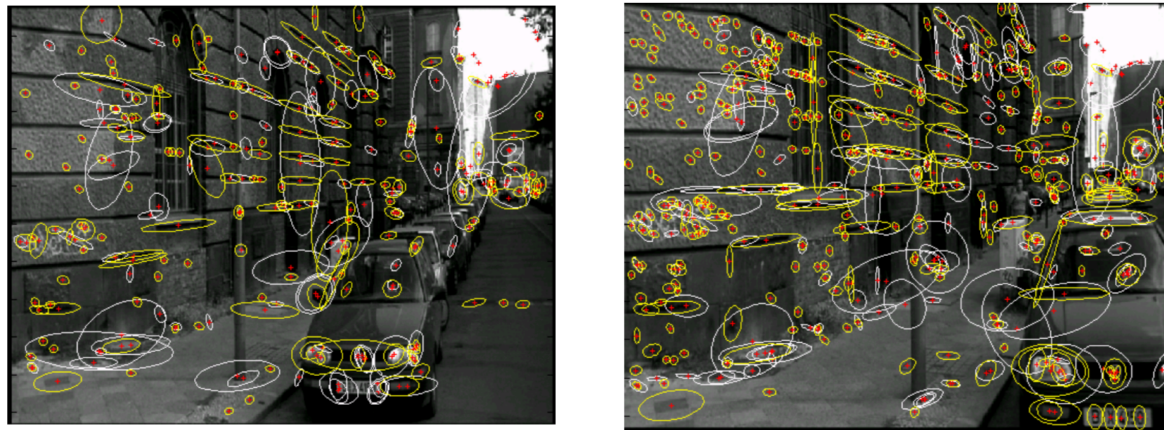
Local representations

- Local descriptors capture the local appearance
- Carefully chosen locations: **repeatability**

Interest point detectors

- Harris, Hessian, Hessian-Affine, MSER, etc.

(more details later)



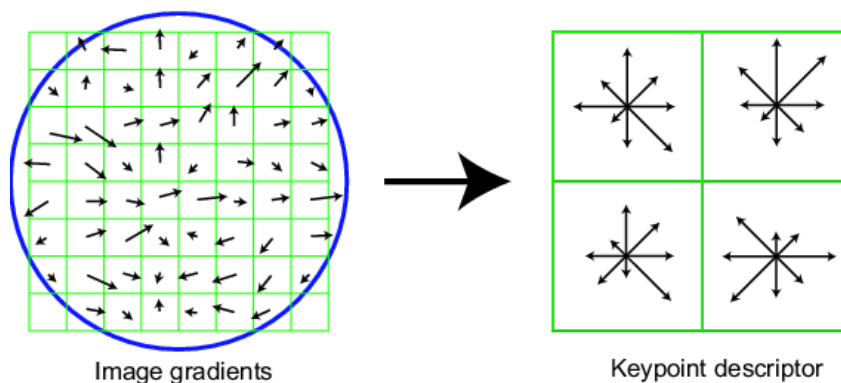
[Harris&Stephens1988, Mikolajczyk&Schmid2002, Matas et al, 2004]
Survey: [Mikolajczyk, IJCV 2005]

Local representations

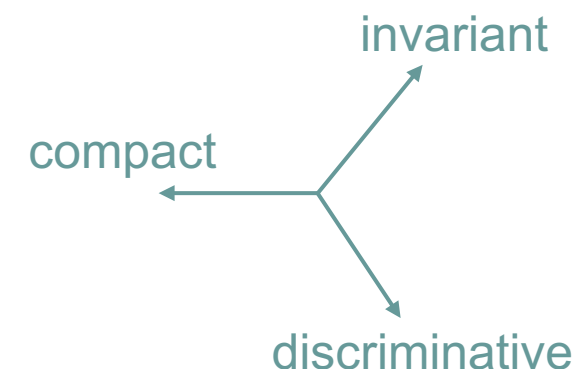
- Local descriptors capture the local appearance: **invariance** and **discriminative**

Local descriptors

- SIFT, SURF, LBP, etc.



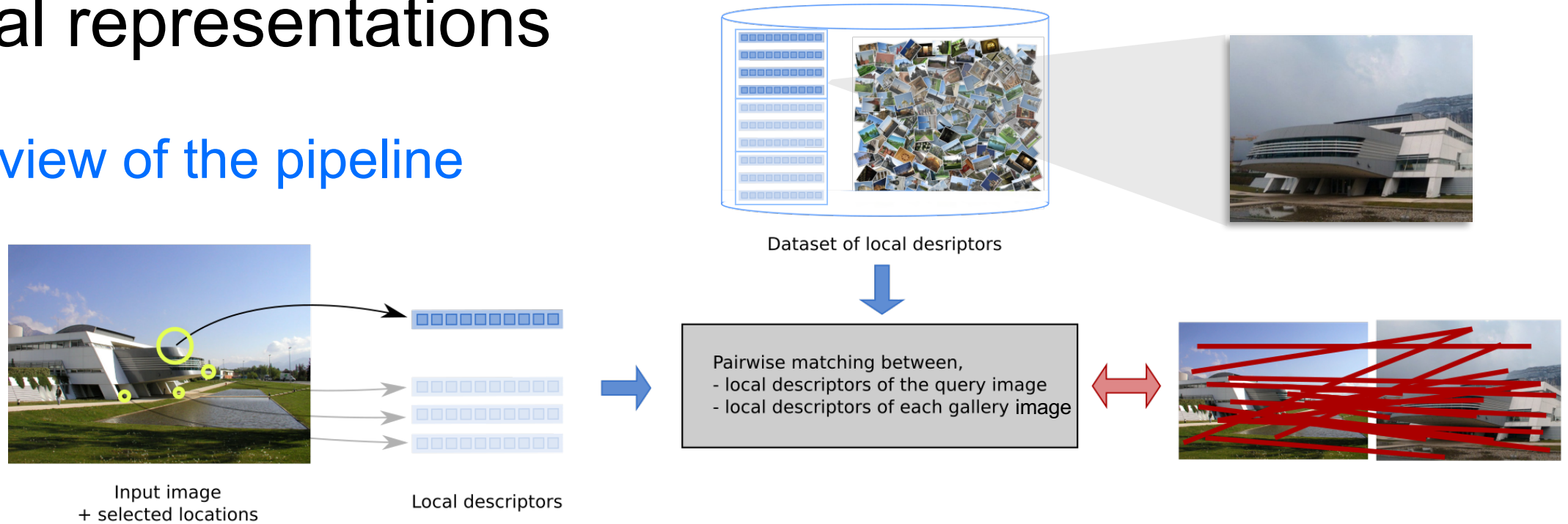
[Lowe IJCV04, Bay et al, 2008, Ojala et al, 2014]



(more details later)

Local representations

Overview of the pipeline



Description locale - plan

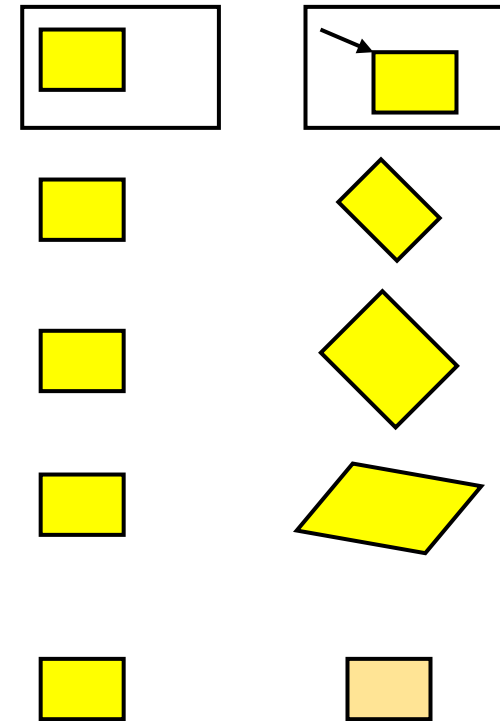
- Extraction de points/régions d'intérêt
- Descripteurs locaux
- Appariement de points

Description locale des images

- Extraction d'informations locales
 - Contours / segments
 - **Points d'intérêt et régions d'intérêt**
 - Régions par segmentation
 - Représentation dense: points sur une grille régulière
- Description locale
 - Dépend du type d'information
 - Contours / segments -> angle, rapport de longueur
 - **Points d'intérêt / grille dense -> niveaux de gris, couleur, histogrammes de gradients**
 - Régions (segmentation) -> distribution de texture et de couleur

Extraction de régions d'intérêt : enjeux

- Extraire des régions qui soient invariantes à de nombreuses transformations, afin d'être « répétables » (*repeatability*)
- Transformations géométriques
 - Translation
 - Rotation
 - Similitude (rotation + changement d'échelle isotrope)
 - Affine (en 3D, texture localement planeaire)
- Transformations photométriques
 - ▶ Changement d'intensité affine ($I \rightarrow a I + b$)



Principaux extracteurs de points/régions d'intérêt

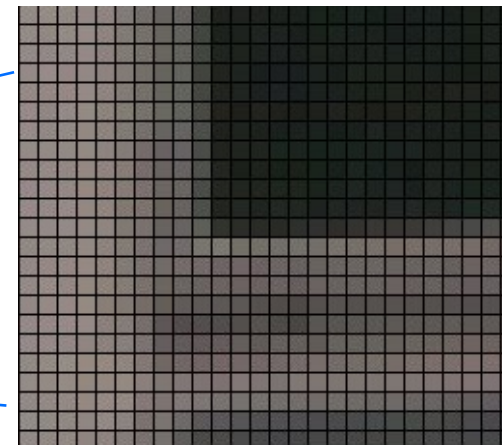
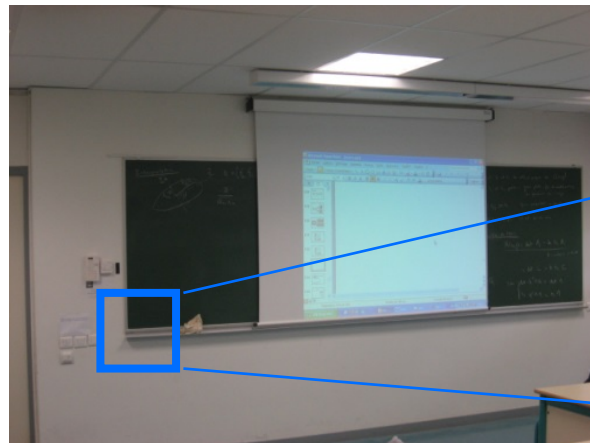
- *Harris*
- *Harris-Affine*
- *Hessian-Affine*
- *Maximally Stable Extremal Regions (MSER)*
- *Salient Region Detector*

[A comparison of affine region detectors » K. Mikolajczyk et al., IJCV 2005]

Détecteur de Harris : Introduction

- Détecter les « coins », qui correspondent à des structures:
 - ▶ répétables de l'image
 - ▶ qui localisent des endroits très discriminants d'une image
- Détecteur de Harris
 - ▶ analyse locale
 - ▶ lorsqu'on est sur un « coin », un déplacement dans n'importe quelle direction produit un fort changement des niveaux de gris

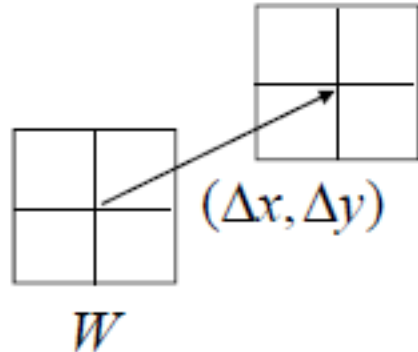
« A Combined Corner and Edge Detector », C. Harris et M. Stephens, 1988



Détecteur de Harris

Auto-correlation function for a point (x, y) and a shift $(\Delta x, \Delta y)$

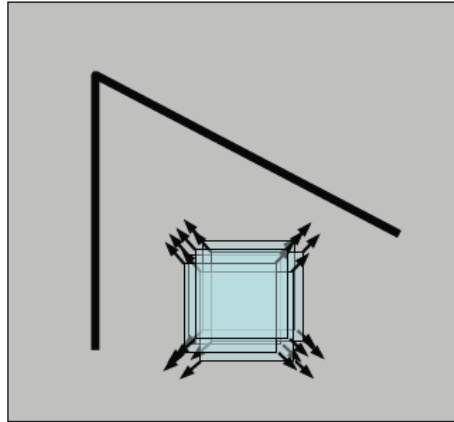
$$A(x, y) = \sum_{(x_k, y_k) \in W(x, y)} (I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y))^2$$



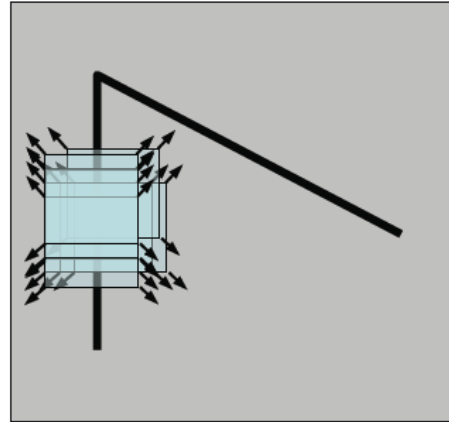
- $A(x, y)$:
 - Faible dans toutes les dimensions -> région uniforme
 - Large dans une direction -> contour
 - Large dans toutes les dimensions -> point d'intérêt

Détecteur de Harris

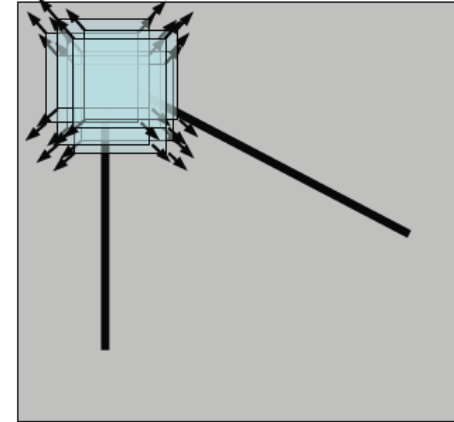
- Illustration:



“flat” region:
no change in
all directions



“edge”:
no change along
the edge direction



“corner”:
significant change
in all directions

Détecteur de Harris

- Comment calculer la fonction d'auto-corrélation ?
- Basé sur une **approximation du premier ordre**

$$I(x_k + \Delta x, y_k + \Delta y) = I(x_k, y_k) + \begin{pmatrix} I_x(x_k, y_k) & I_y(x_k, y_k) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

$$\begin{aligned} A(x, y) &= \sum_{(x_k, y_k) \in W(x, y)} (I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y))^2 \\ &= \sum_{(x_k, y_k) \in W} \left(\begin{pmatrix} I_x(x_k, y_k) & I_y(x_k, y_k) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right)^2 \end{aligned}$$

Détecteur de Harris

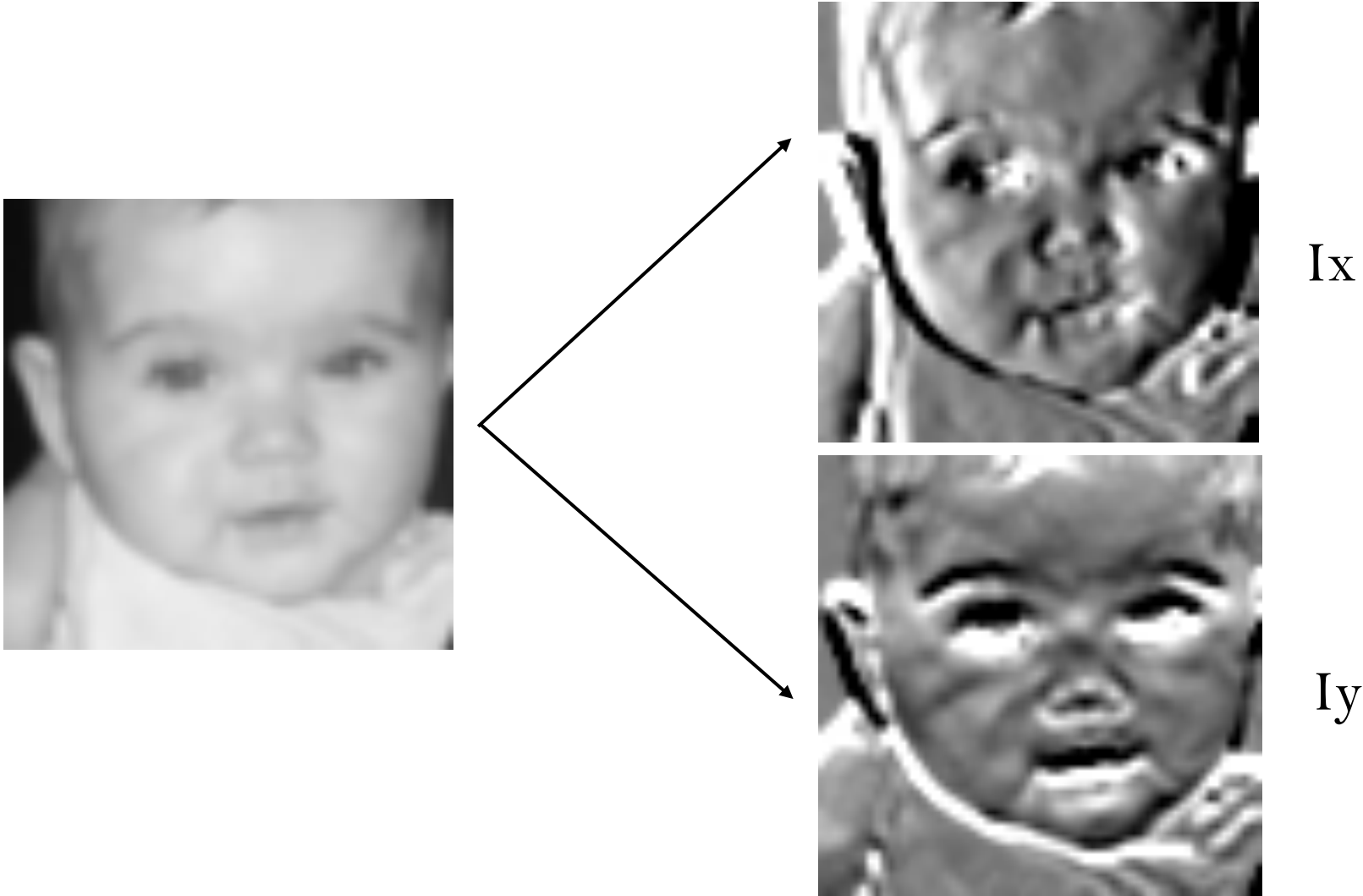
- Basé sur une approximation du premier ordre

$$= \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \begin{bmatrix} \sum_{(x_k, y_k) \in W} (I_x(x_k, y_k))^2 & \sum_{(x_k, y_k) \in W} I_x(x_k, y_k) I_y(x_k, y_k) \\ \sum_{(x_k, y_k) \in W} I_x(x_k, y_k) I_y(x_k, y_k) & \sum_{(x_k, y_k) \in W} (I_y(x_k, y_k))^2 \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

Auto-correlation matrix

Harris : extraction du patch et calcul des gradients

- Calcul des gradients
 - On calcule indépendamment le gradient dans les deux dimensions x et y



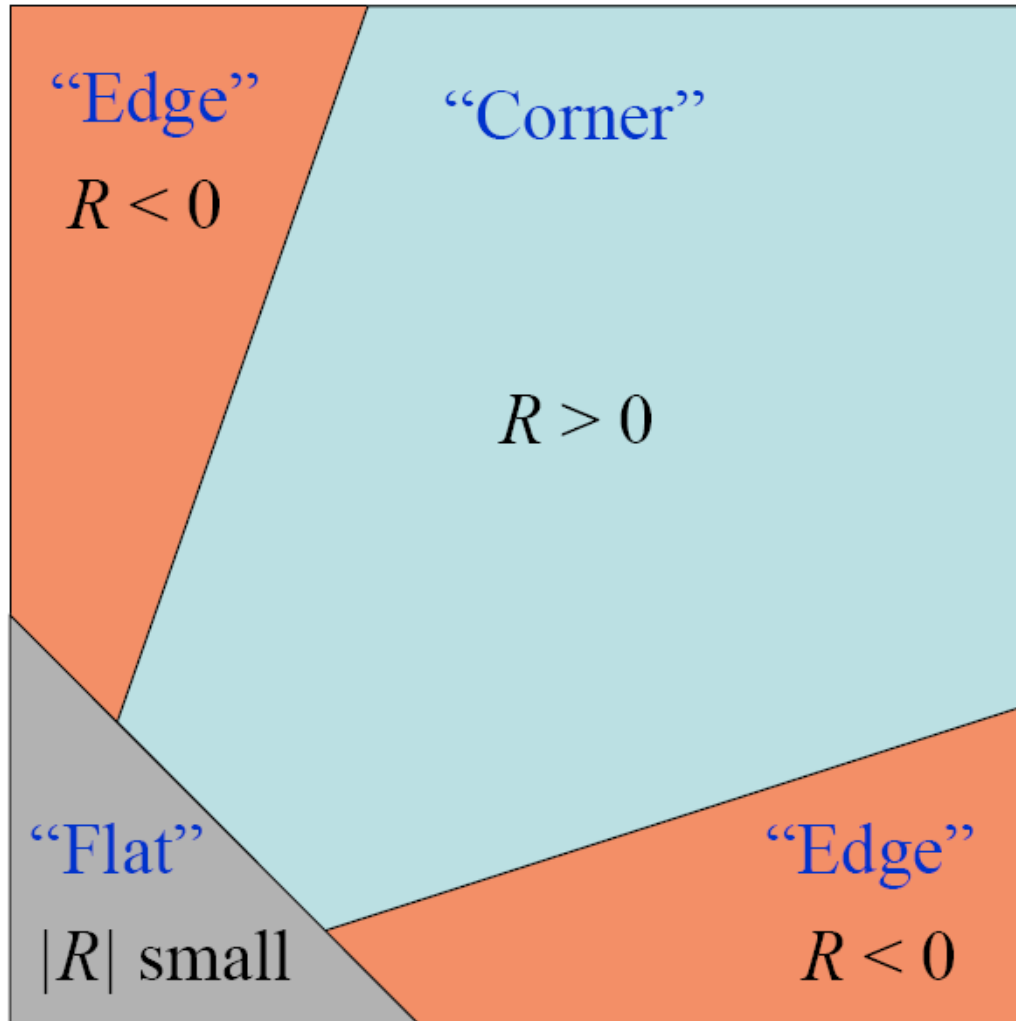
Harris : deuxième étape

- Calcul des gradients
 - Application d'un filtre gaussien pour « régulariser » (*smoothing*) et être moins sensible au bruit
 - Propriété intéressante du filtre dérivé de la Gaussienne



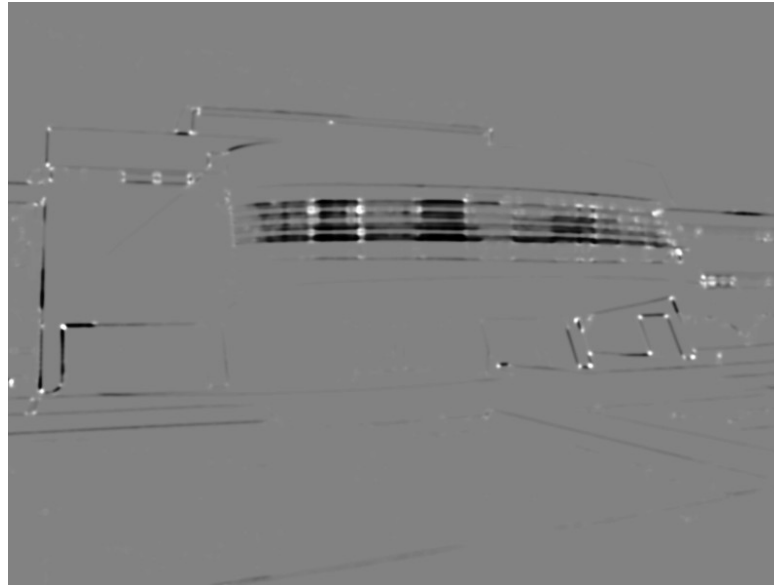
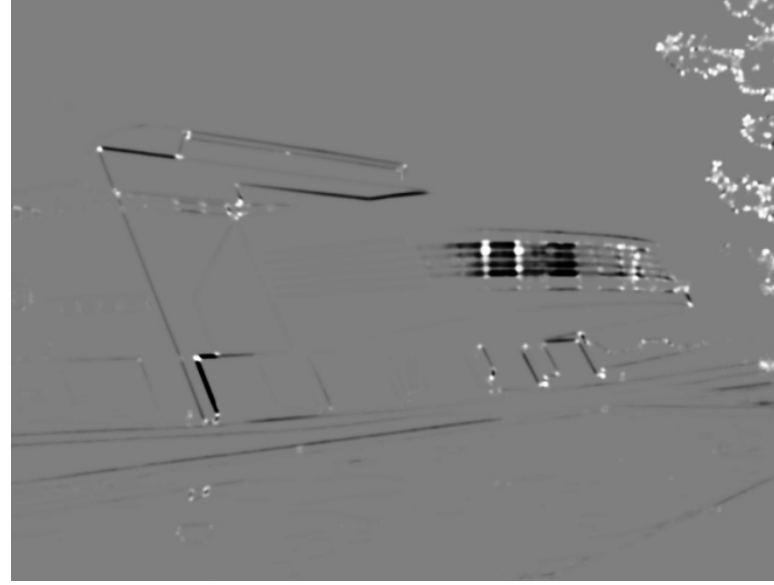
Harris : classification des régions : utilisation d'un seuil

$$R = \det(A) - \alpha \operatorname{trace}(A)^2 = \lambda_1 \lambda_2 - \alpha (\lambda_1 + \lambda_2)^2$$



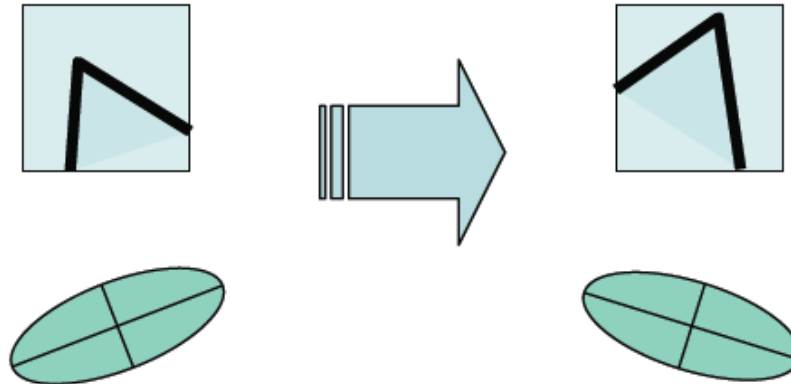
- Détection de points d'intérêt
 - Application d'un seuil (absolu, relatif, nombre de points)
 - **Extraction des maxima locaux**

Harris : exemple



Harris : Invariance

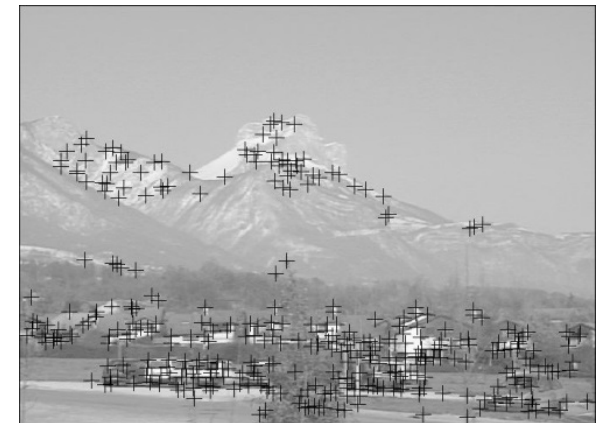
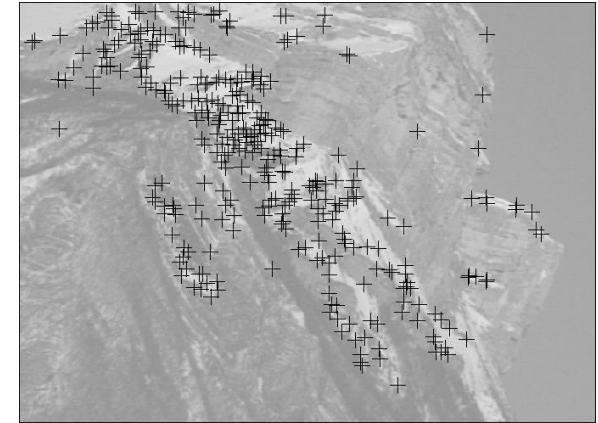
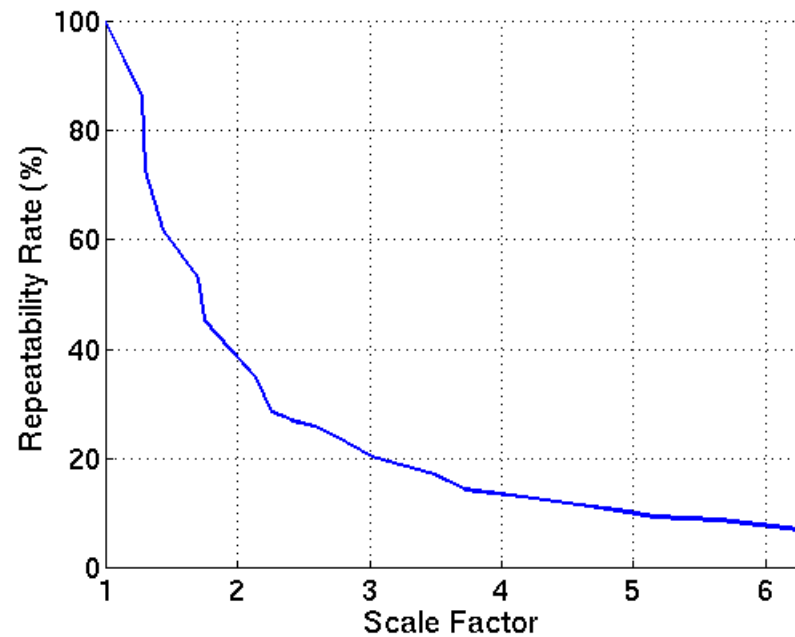
- Invariance à la translation : inhérente à la sélection des maxima locaux
- Invariance aux transformations affines d'intensité
- Invariance à la rotation
 - ▶ l'ellipse tourne
 - ▶ mais la forme reste identique
 - ▶ les valeurs propres restent identiques



- Mais pas d'invariance à des changements d'échelle

Harris : Invariance

- Pas d'invariance aux changements d'échelle
- Taux de répétabilité



Détecteurs Hessien & DoG

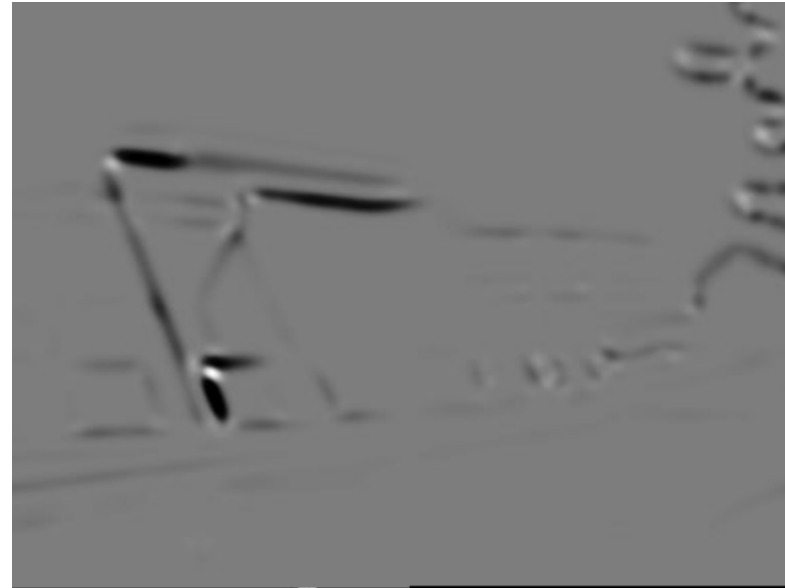
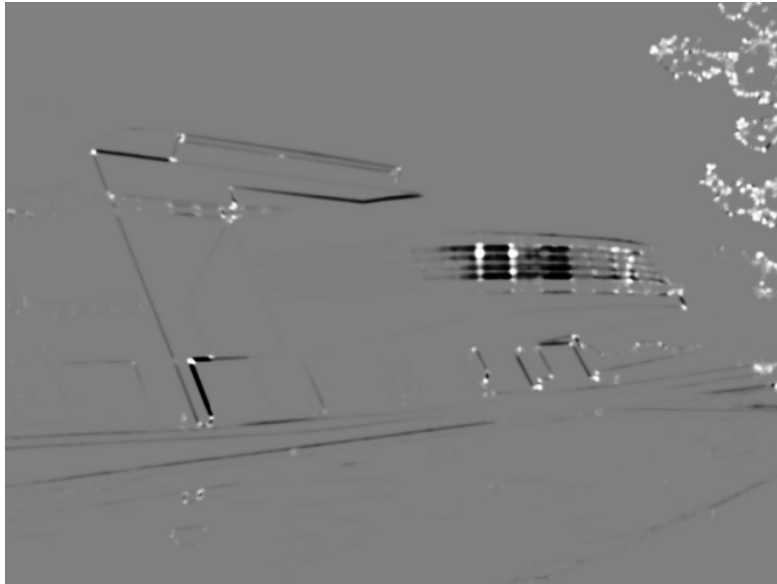
- Analogue au détecteur de Harris
- Les valeurs propres de la matrice des dérivées secondes (Hessienne) caractérisent la forme locale du voisinage d'un point.
- Max sur le déterminant

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

- Forte réponse sur les « blobs » et les crêtes
 - Autre détecteur de blob: le détecteur DoG (*Difference of Gradients*)
- Les détecteurs de **coins** et de **blobs** répondent à des structures différentes

Invariance à l'échelle (début)

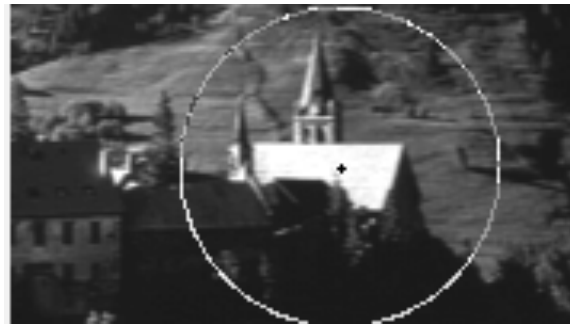
- Comment obtenir un détecteur invariant au changement d'échelle
- Echelle de calcul de Harris/Hessien/DoG
 - ▶ Déterminée par la taille des filtres de dérivation : σ des filtres gaussiens
 - ▶ Réagit à une certaine taille de détails / une certaine plage de fréquences spatiales



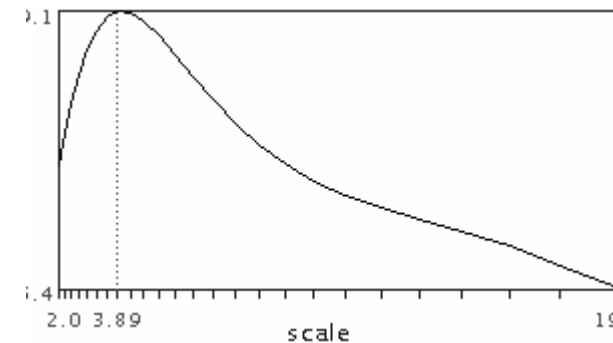
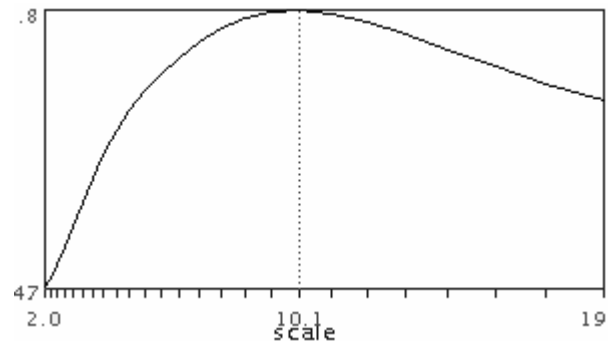
Invariance à l'échelle (suite)

- Noyau (*kernel*) : Laplacien ou DoG
- Sélection d'une échelle caractéristique :
 - ▶ prendre les valeurs extrêmes d'une fonction f liée à l'échelle
 - ▶ Une bonne fonction f présente des pics marqués

$$\nabla \cdot \nabla f = \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$



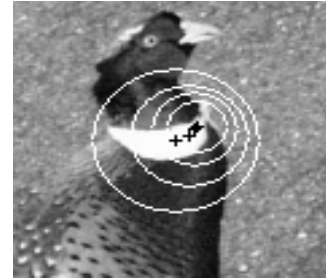
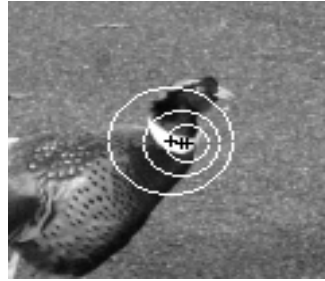
Laplacien



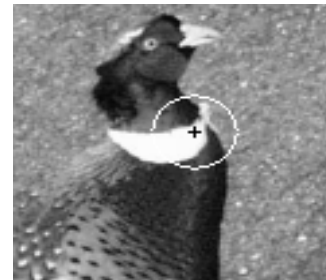
invariance de l'échelle caractéristique

Invariance à l'échelle (suite)

- Détecteur de Harris-Laplace
 - ▶ Sélectionner le maximum de Harris en espace pour plusieurs échelles

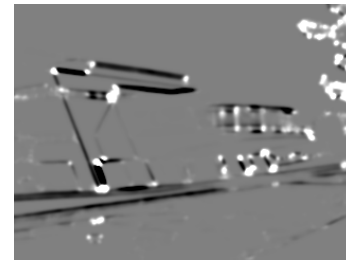
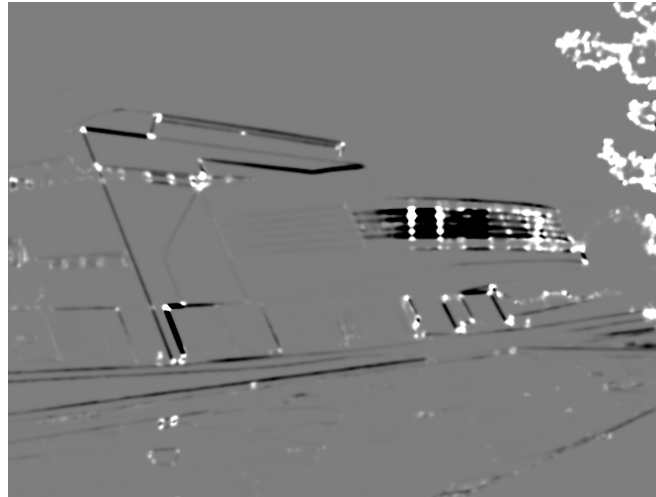
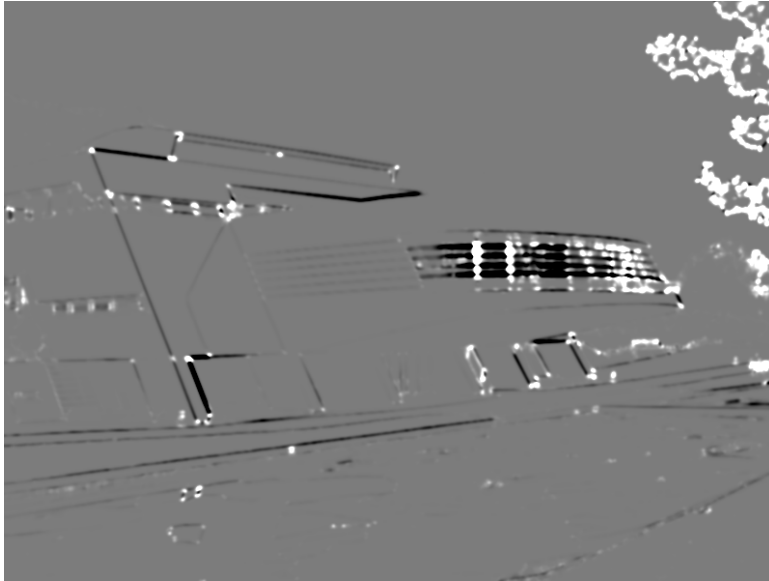


- ▶ Sélection des points à leur échelle caractéristique avec le Laplacien



Invariance à l'échelle (fin)

- Multi-échelle en pratique:
 - ▶ pyramide d'images
 - ▶ Gaussiennes de taille constante

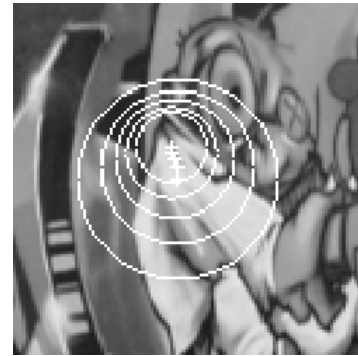
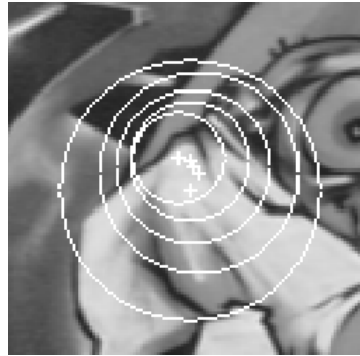


Harris-Affine et Hessian-Affine (début)

- Détecteur de région de l'état de l'art
- Algorithme : estimation itérative des paramètres
 - ▶ localisation : Harris
 - ▶ échelle : sélection automatique avec le Laplacien
 - ▶ **voisinage affine : normalisation avec la matrice des seconds moments**→ jusqu'à convergence

Harris-Affine et Hessian-Affine : pour résumer

- Initialisation avec des points d'intérêt multi-échelle



- Estimation itérative de la localisation, de l'échelle, du voisinage

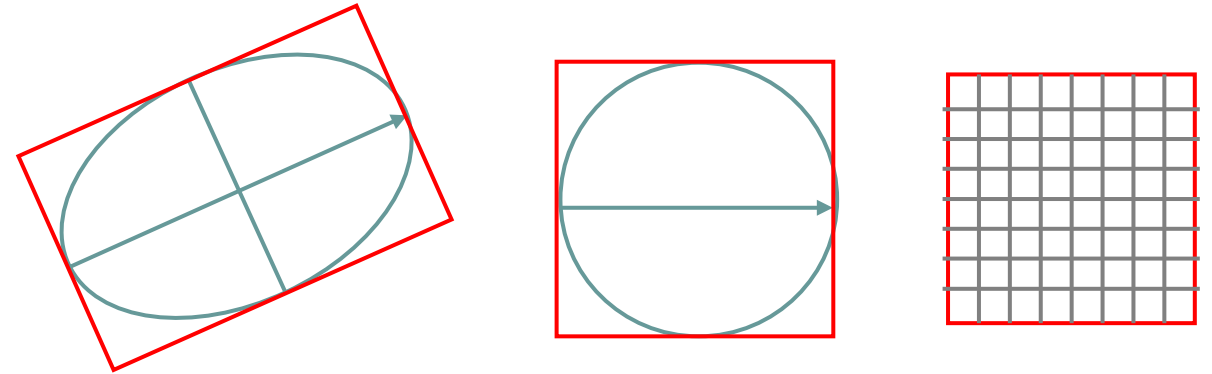


Description locale : plan

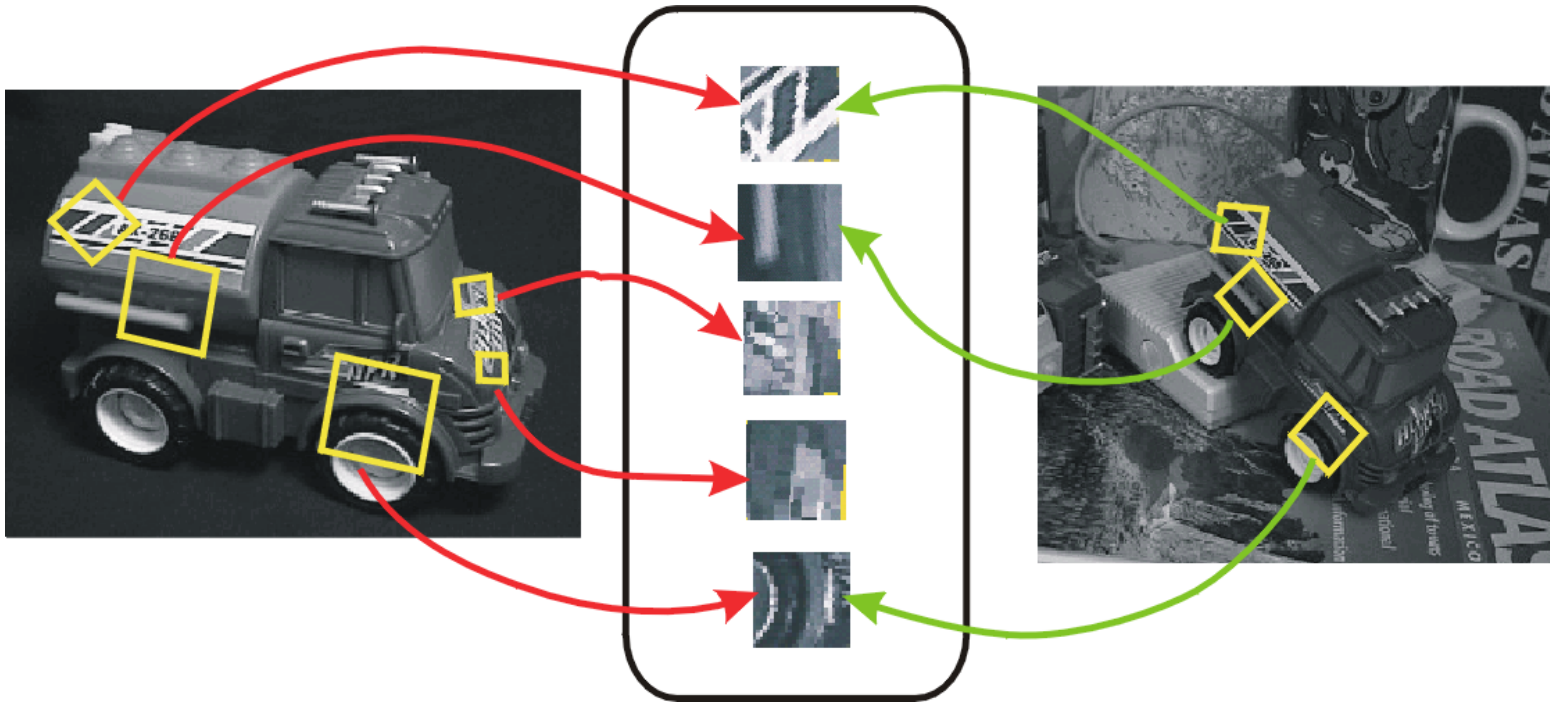
- Extraction de points/régions d'intérêt
- Descripteurs locaux
- Appariement de points

Et une fois qu'on a défini l'ellipse ?

- Synthèse du *patch* (imagelette)
 - ▶ On calcule un patch de taille donnée

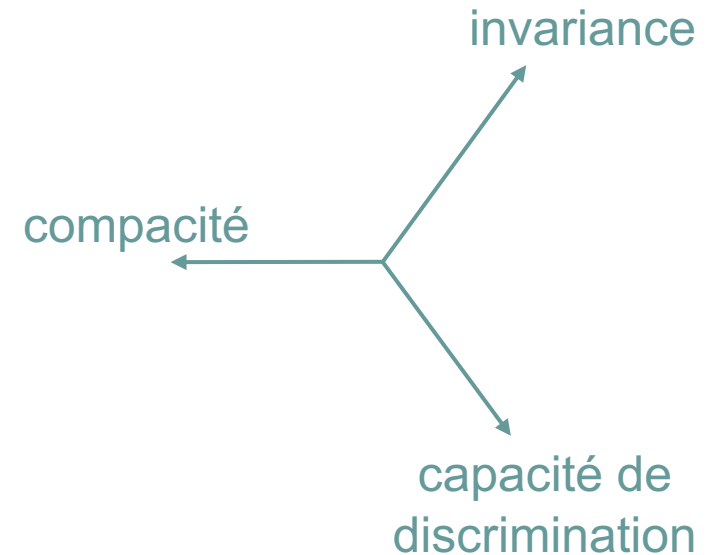


- Tailles typiques de patch : 21x21, 64x64



Objectif

- En utilisant un détecteur, on est capable de détecter des zones robustes (et informatives) de l'image
- Mais à ce stade, ces zones ne peuvent pas être comparées entre elles
- Qu'est qu'un descripteur ?
 - ▶ une représentation de la zone d'intérêt
 - ▶ sous la forme d'un vecteur
 - ▶ qui appartient à un espace *muni d'une distance*
- Qu'est qu'un bon descripteur ?
 - ▶ invariant
 - ▶ discriminant
 - ▶ compact
- Remarque : ces objectifs sont contradictoires

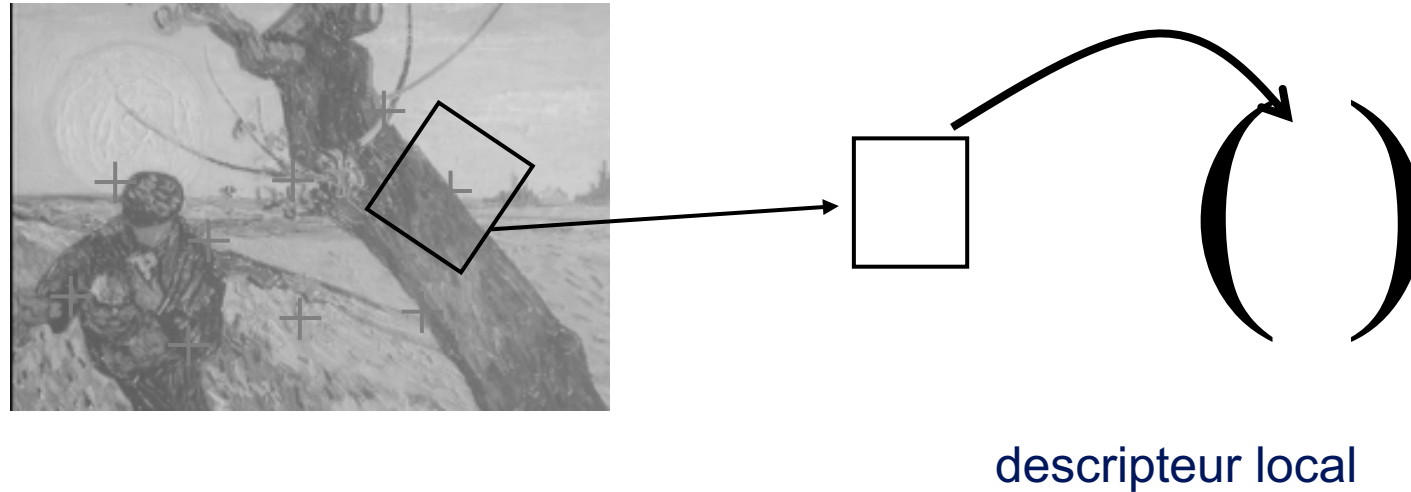


Descripteurs

- Descripteurs basiques
- État de l'art : le descripteur SIFT
- Descripteur CS-LBP

Descripteurs de pixel

- On range les pixels au voisinage du point dans un vecteur

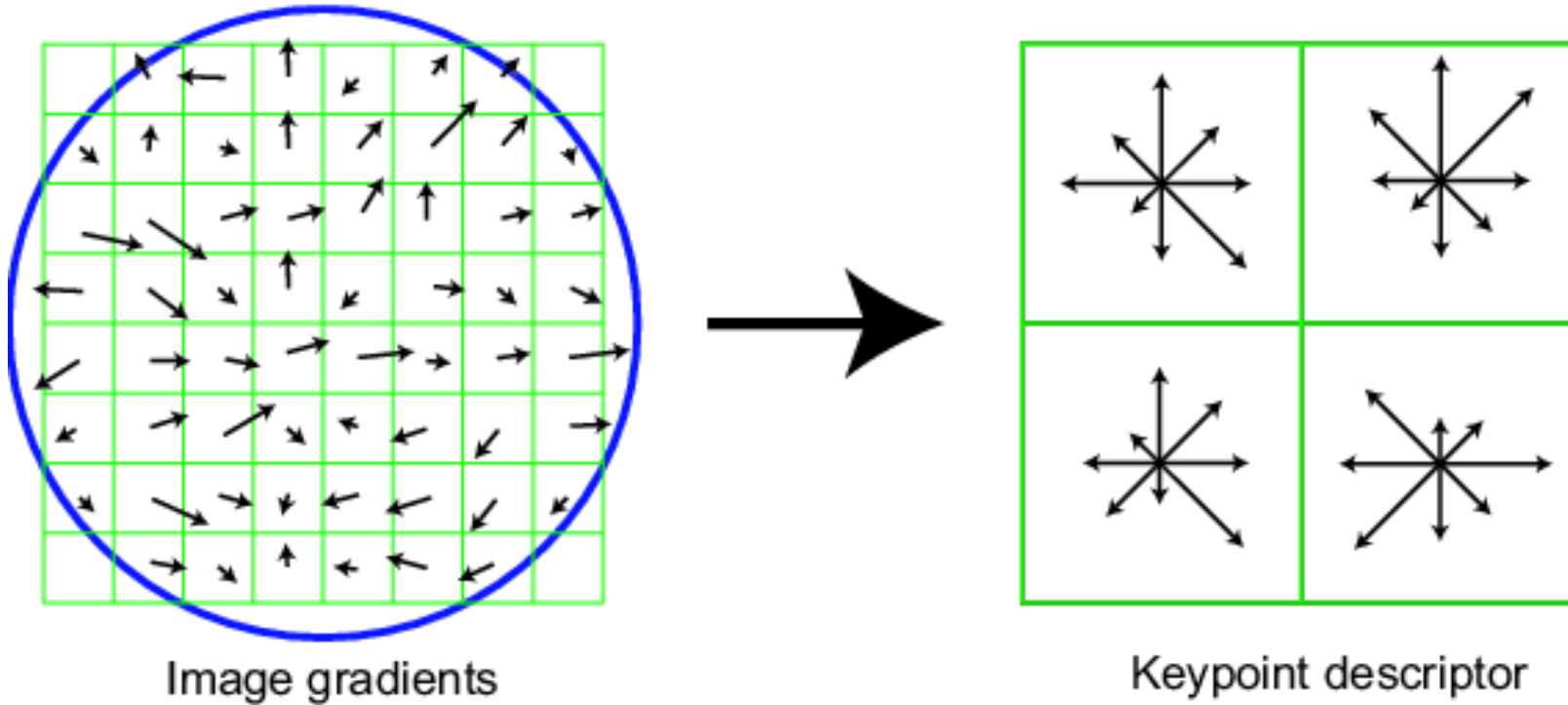


- Calcul simple
- Volumineux (patch de 32×32 = vecteur de 1024 éléments)
- Invariance possible aux transformations affines de luminosité
- Faible « baseline »

SIFT (Scale invariant feature transform)

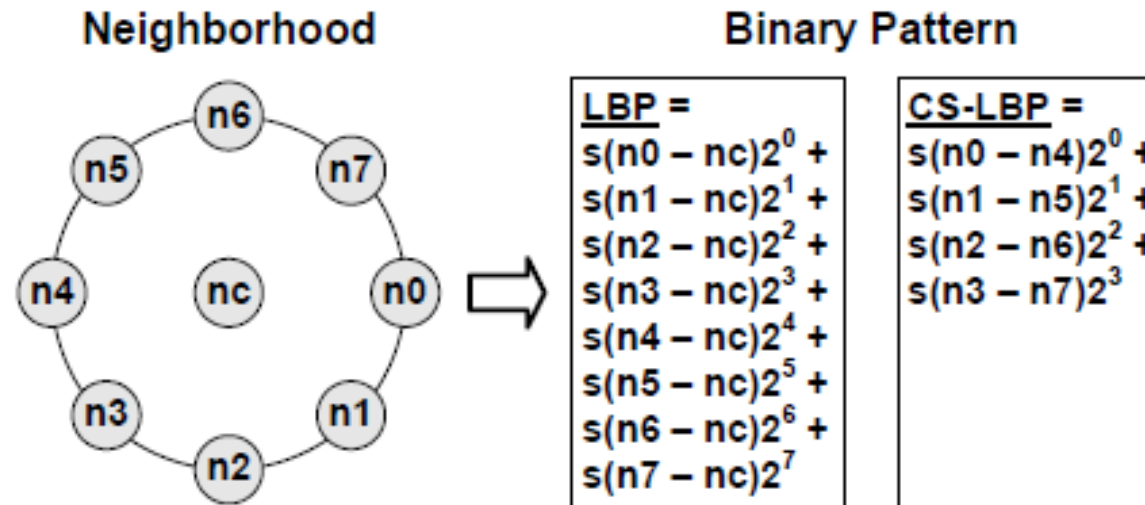
- Calculé sur un patch déjà normalisé : échelle (ou affine)
- Possibilité de normaliser également en orientation (orientation dominante)

Description détaillée au tableau



CS-LBP (*Center-symmetric local binary pattern*)

- CS-LBP principe



- nombreuses autres variantes de SIFT:
 - ▶ HOG (*generalization*)
 - ▶ GLOH (*log-polar bins*)
 - ▶ SURF (*approximation, Blob detector, HAAR wavelets*)

Navneet Dalal et Bill Triggs, "Histograms of Oriented Gradients for Human Detection". CVPR 2005

M. Heikkila, M. Pietikainen, C. Schmid, "Description of interest regions with local binary patterns"

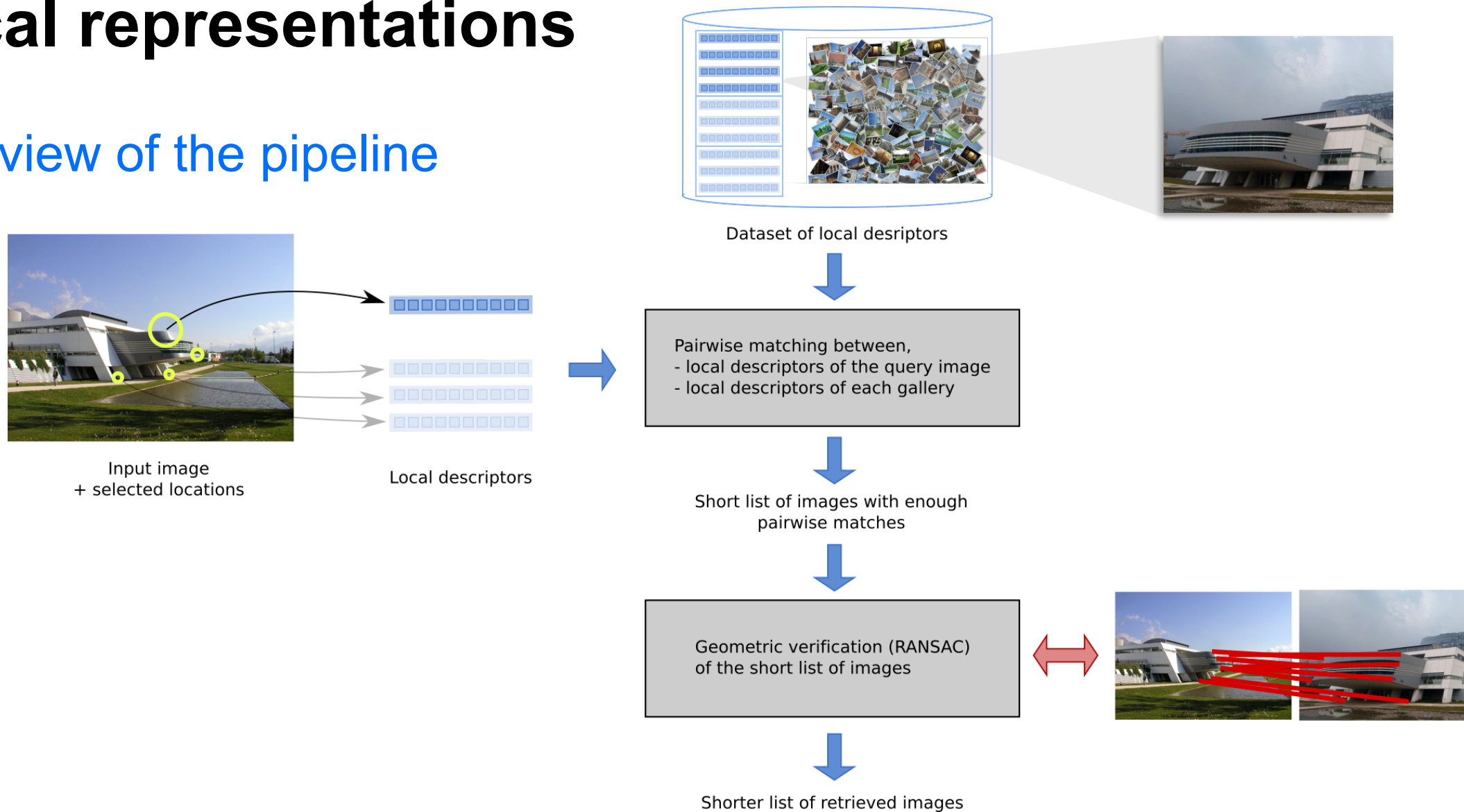
E. Tola, V. Lepetit, P. Fua, "A fast local descriptor for dense matching". CVPR 2008

Descripteurs : conclusion

- **Invariances** : force des méthodes génériques
 - ▶ normalisation affine de la région vers un patch de taille constante en utilisant un détecteur affine invariant
 - ▶ normalisation affine en luminosité avant le calcul du descripteur pour garantir l'invariance affine
- Utilisation d'un descripteur de relativement grande dimension (128 pour SIFT) pour obtenir une bonne capacité de **discrimination**

Local representations

Overview of the pipeline



Description locale : plan

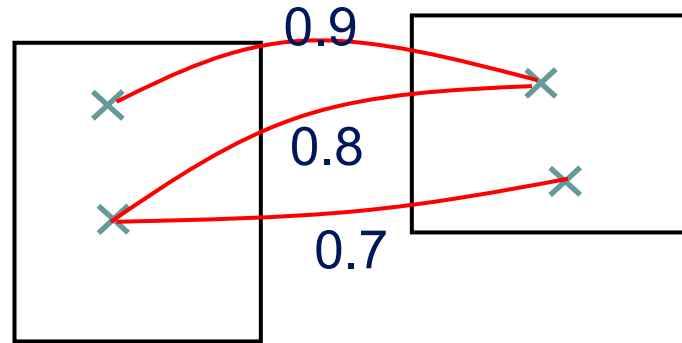
- Extraction de points/régions d'intérêt
- Descripteurs locaux
- Appariement de points

Appariement de points

- Problème : étant donné des points ou régions d'intérêt et les descripteurs associés, comment apparier de manière robuste les points de deux images ?
 - Dans ce contexte: robuste = robuste au bruit = au mauvais appariements locaux
- Dans la suite: sélection des meilleurs couples de points

Sélection des meilleurs couples

- Hypothèse : pour tout couple de points (p_{1i}, p_{2j}) , avec p_{1i} dans I_1 et p_{2j} dans I_2 , on a une mesure de similarité s_{ij} entre régions, en général par calcul d'une similarité entre leur descripteur
- Remarque: ambiguïté

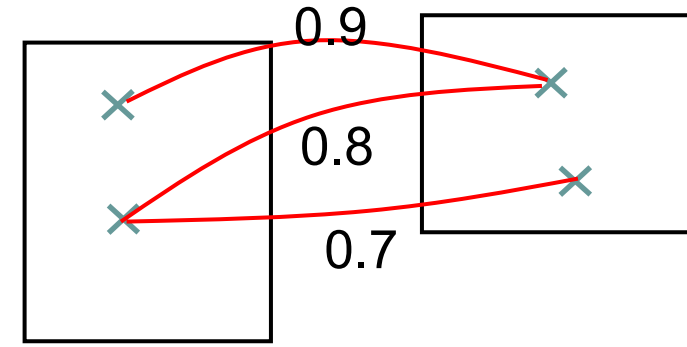


- Différentes stratégies:
 - ▶ *Winner-takes-all* (symétrique ou à sens unique)
 - ▶ Filtrage ultérieur par géométrie

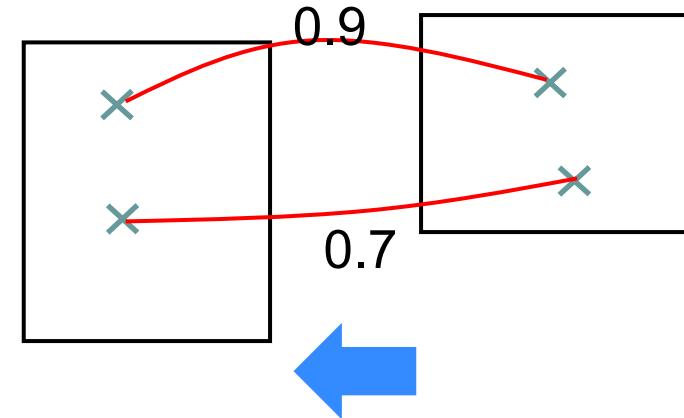
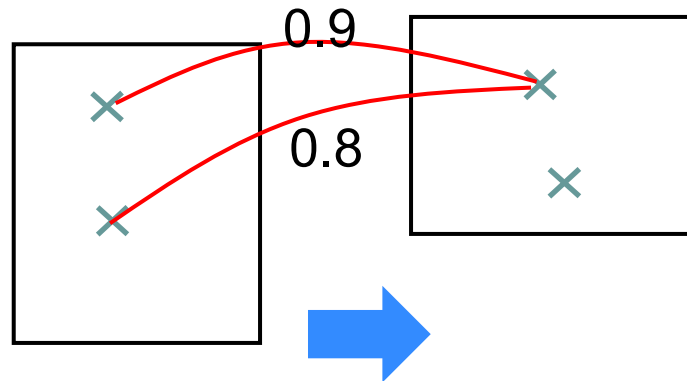
Sélection des meilleurs couples

Différentes stratégies:

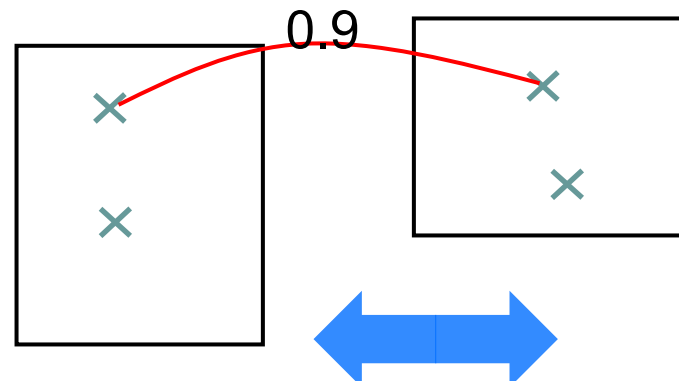
- **Winner-takes-all**



A sens unique



Symétrique



Problème : appariements entre deux images



Vérification géométrique par mise en correspondance d'images

Différentes stratégies:

- ▶ **Filtrage ultérieur par géométrie**

Deux méthodes classiques

- RANSAC
- Transformée de Hough

Vérification par un modèle géométrique

Pour une paire d'images I, I'

- Il existe un vecteur de paramètres p
tel que 2 points en correspondance (x, x') vérifient $F(x, x', p) = 0$
- le plus souvent $x' = T(x, p)$ avec T transformation
- En indexation:
 - ▶ si les images sont en correspondance, alors il existe p tel que “suffisamment” de points (x, x') vérifient $F(x, x', p) = 0$

Idée intuitive de l'algorithme:

1. estimation de p
2. nombre de points qui vérifient l'équation avec le meilleur p = mesure de ressemblance entre images

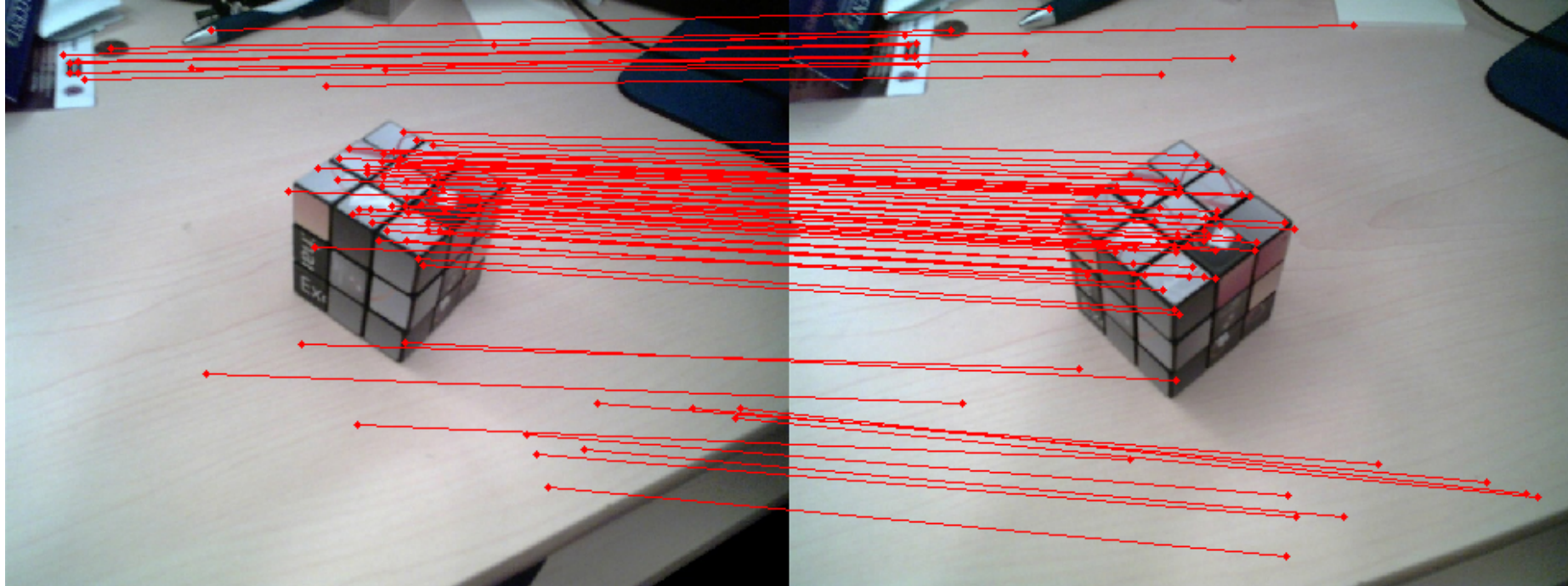
RANSAC

RANSAC (*RAN*dOm *SA*mple *C*onsensus)

- Algorithme
 - Entrée: un ensemble d'appariements locaux entre deux images
 - Sortie: paramètres de la transformation estimée entre les deux images
- Pour *try* dans $1..K$
 - Sélectionner aléatoirement un ensemble d'appariements (*seed*)
 - Estimer les paramètres de la transformation entre les deux images en utilisant seulement cet ensemble *seed*
 - Trouver le nombre d'appariements qui correspondent à la transformation estimée (*inliers*)
 - Si ce nombre est suffisamment grand
 - Reestimer les paramètres avec tous les *inliers*
- Garder les paramètres de la transformation qui a le plus d'*inliers*

Problème : appariements entre deux images

après RANSAC

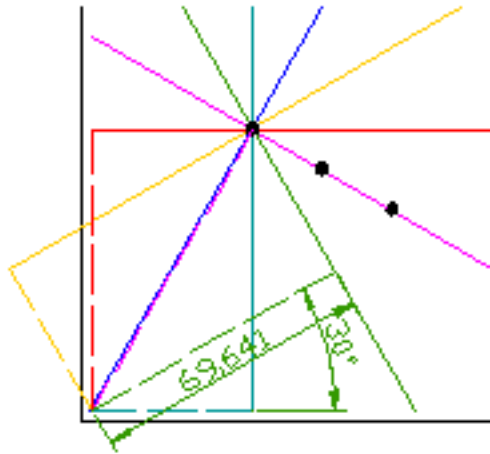


Transformée de Hough

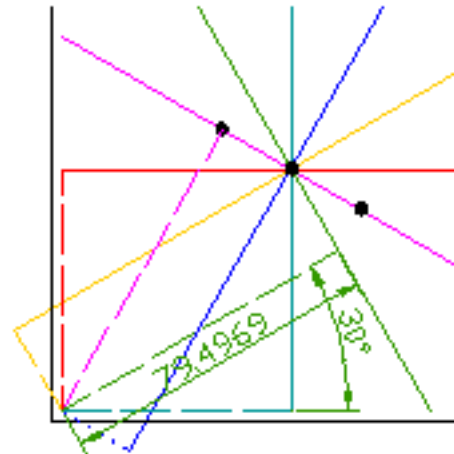
- Utilisée initialement pour détecter des segments/droites à partir de points dans les images
 - ▶ Chercher les droites possibles dans un espace de paramètres (espace de Hough)
- Principe :
 - ▶ discrétiser l'espace de paramètres en cellules (*bins*)
 - ▶ Pour chaque observation, incrémenter la(s) cellule(s) cohérente(s) avec l'observation
 - ▶ Résultat = paramètres correspondants à la cellule la plus peuplée
 - ▶ affinage possible avec *Mean-shift*
- Difficulté:
 - Dépendance à la paramétrisation et à la quantification
 - Dans le cas de la vérification géométrique, le nombre de degrés de liberté possibles doit être petit !!

Transformée de Hough

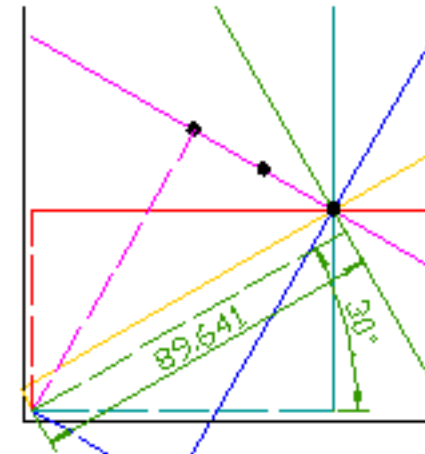
- Illustration (wikipedia)



Angle	Dist.
0	40
30	69.6
60	81.2
90	70
120	40.6
150	0.4



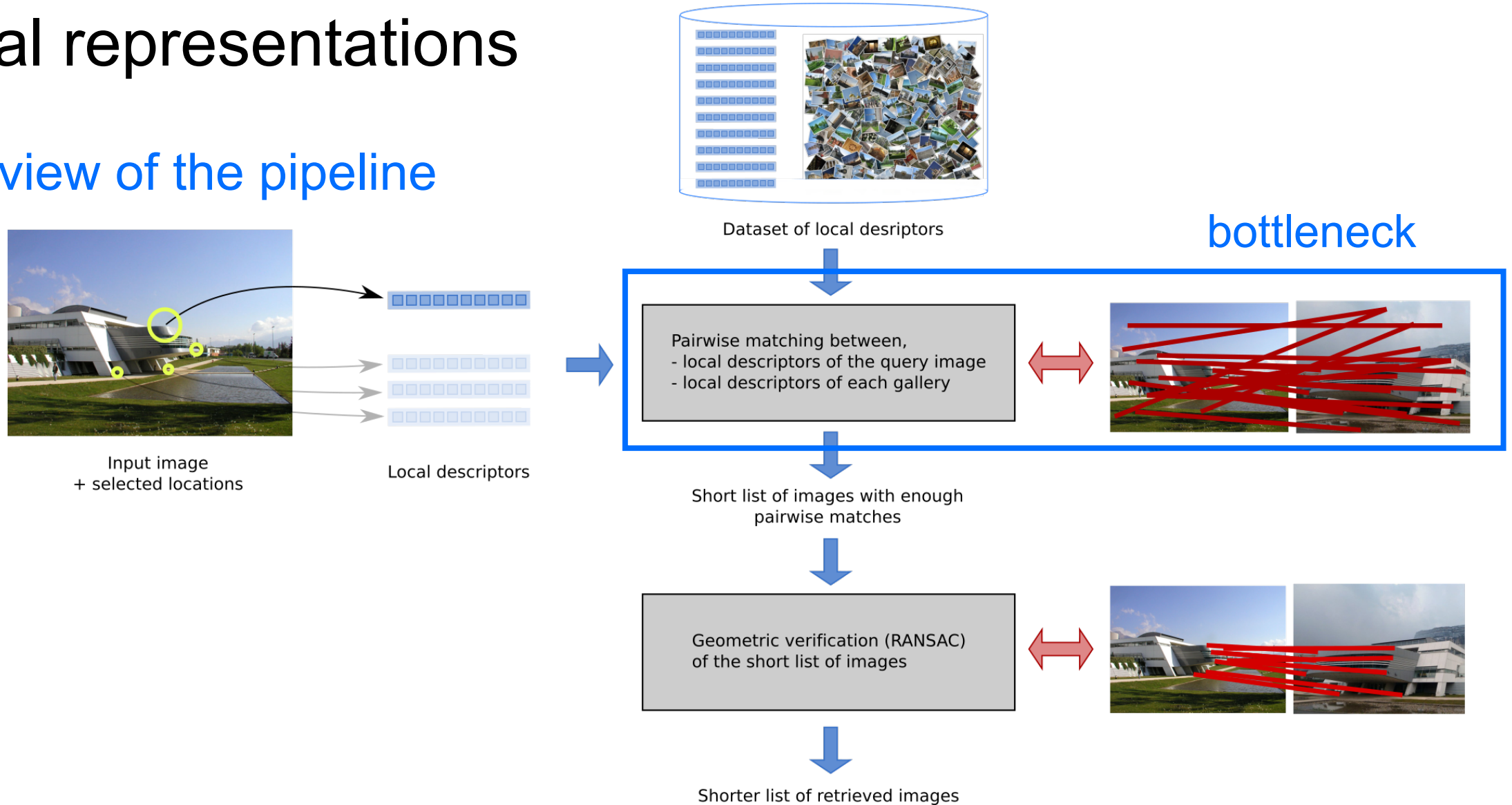
Angle	Dist.
0	57.1
30	79.5
60	80.5
90	60
120	23.4
150	-19.5



Angle	Dist.
0	74.6
30	89.6
60	80.6
90	50
120	6.0
150	-39.6

Local representations

Overview of the pipeline



Scaling the first selection process

1) Efficient approximate nearest neighbor search on local descriptors

- Randomized K-d trees and variants
 - Priority queue to examine most promising first
- Locality-Sensitive Hashing (LSH)
 - Randomized hashing technique
 - Data-dependent variants: spectral hashing, semantic hashing

[Silpa-Anan & Hartley 2008,
Chum et al 2008, Muja & Lowe. 2009]

[Indik & Motwani, 1998, Weiss et al NIPS08,
Salakhutdinov & Hinton, SIGIR07]

Scaling the first selection process

1) Efficient approximate nearest neighbor search on local descriptors

- Randomized K-d trees and variants
 - Priority queue to examine most promising first
- Locality-Sensitive Hashing (LSH)
 - Randomized hashing technique
 - Data-dependent variants: spectral hashing, semantic hashing

[Silpa-Anan & Hartley 2008,
Chum et al 2008, Muja & Lowe. 2009]

[Indik & Motwani, 1998, Weiss et al NIPS08,
Salakhutdinov & Hinton, SIGIR07]

2) Quantize local descriptor space into a visual codebook

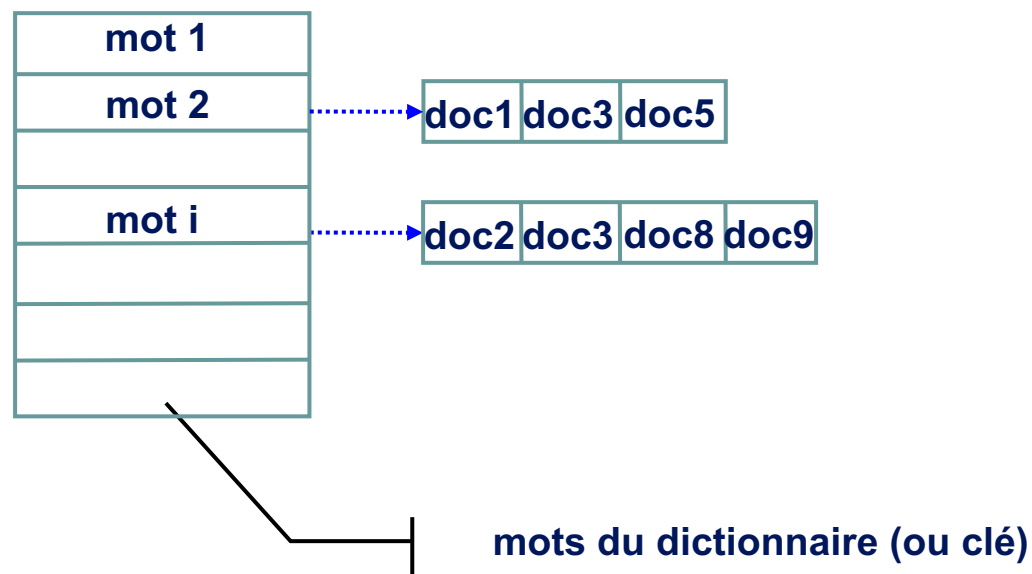
- Uses **inverted files** to store the descriptors
- Efficient image matching

Plan de la suite

- Concept de fichier inversé - motivation
- Exemple d'utilisation d'un fichier inversé dans le cas de la recherche de texte
- Comment créer un vocabulaire visuel pour utiliser cette technique en conjonction avec des représentations locales d'images
- Application d'un fichier inversé au cas de la recherche d'images

Fichier inversé (*inverted file*)

- Cette structure liste des éléments qui ont une valeur donnée pour un attribut
- Bases de données: utilisé pour les clés secondaires (doublons attendus)
- Exemple courant d'utilisation : requêtes sur documents textuels (mails, ...)



- La requête consiste à récupérer la liste des documents contenant le mot
 - ▶ coût proportionnel au nombre de documents à récupérer

Recherche de documents textuels (1)

- Modèle vectoriel
 - ▶ on définit un vocabulaire de mots de taille d (suppose un vocabulaire fini)
 - ▶ un document texte est représenté par un vecteur
 $f = (f_1, \dots, f_i, \dots, f_d) \in \mathbb{R}^d$
 - ▶ chaque dimension i correspond à un mot du vocabulaire
 - ▶ f_i = fréquence du mot dans le document
- en pratique, on ne garde que les mots discriminants dans le vocabulaire
 - “le”, “la”, “est”, “a”, etc, sont supprimés, car peu discriminants
- Ces vecteurs sont creux
 - ▶ Le vocabulaire est grand par rapport au nombre de mots utilisés

Recherche de documents textuels (2)

- Exemple

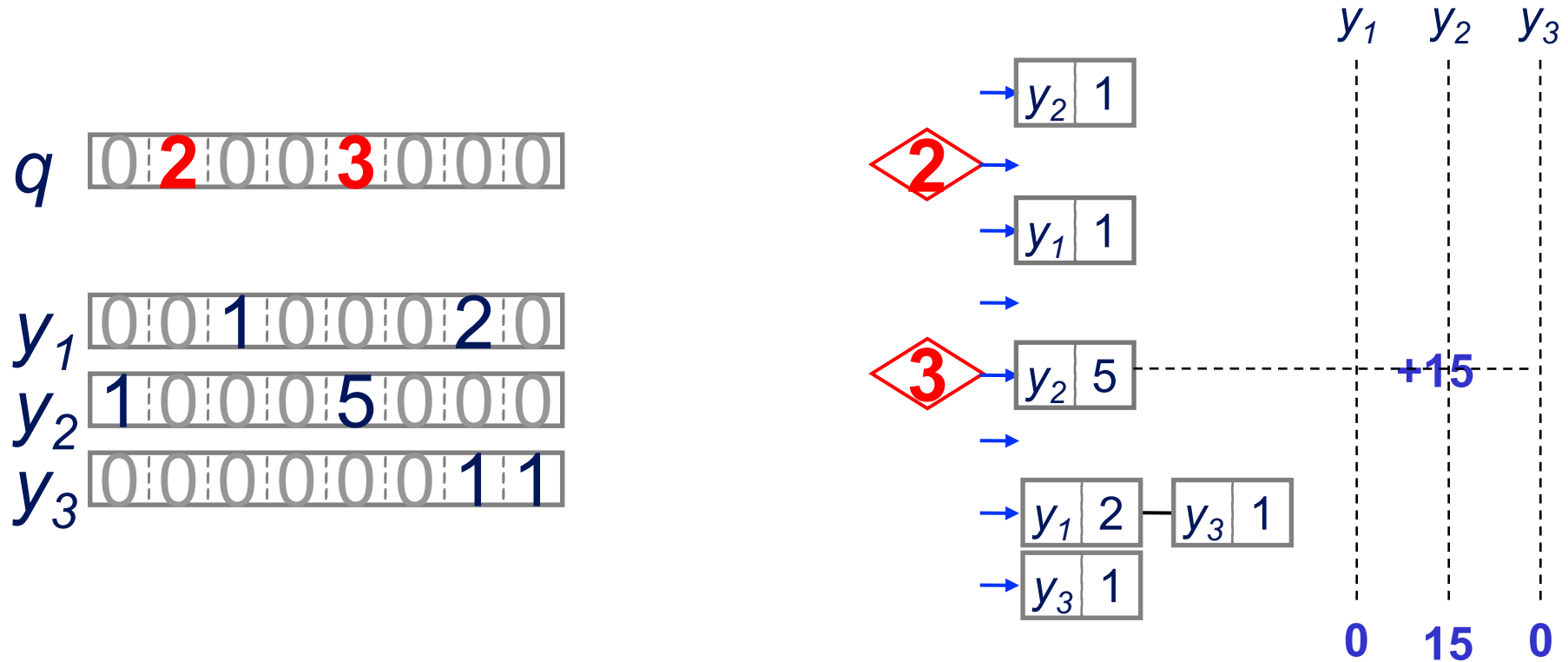
- ▶ vocabulaire = {"vélo", "voiture", "déplace", "travail", "école", "Grenoble"}
- ▶ espace de représentation: $\mathbb{R}^6$
- ▶ "Grenoble est une belle ville. Je me déplace à vélo dans Grenoble"

$$f=(1/4,0,1/4,0,0,1/2)^t$$

- Recherche de document = trouver les vecteurs les plus proches
 - ▶ au sens d'une mesure de (dis-)similarité, en particulier le produit scalaire

Fichier inversé : distances entre vecteurs creux

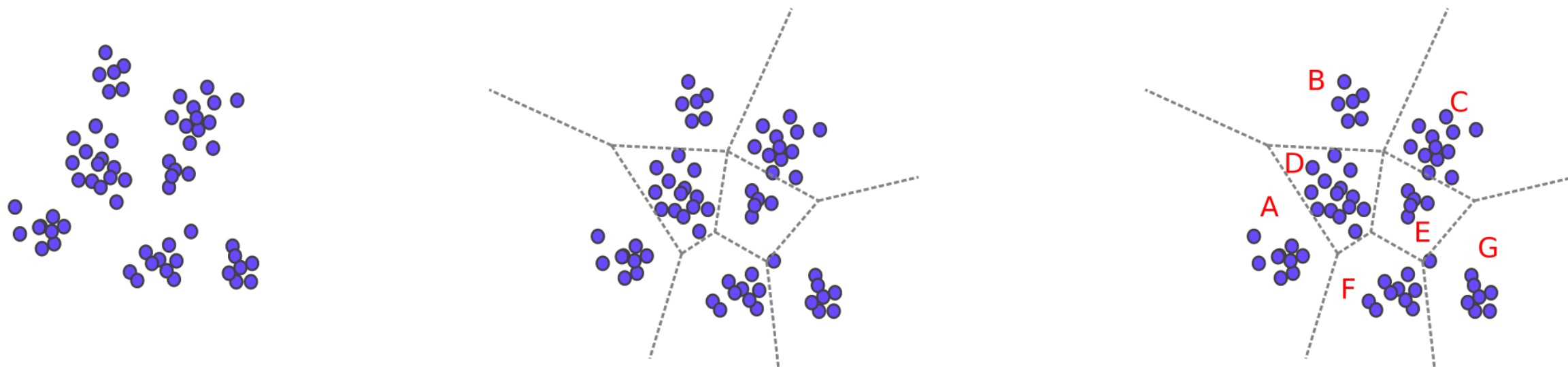
- La requête q et les éléments de la base Y sont des vecteurs creux
- Fichier inversé: calcul efficace du produit scalaire (en fait, toute distance L_p)
- Exemple pour le produit scalaire



Quantization: principle

Principle:

- Discretize the local descriptor space
- 2 descriptors match i.i.f. they fall in the same bin
- this creates a visual codebook, similar to the textual codebook seen before

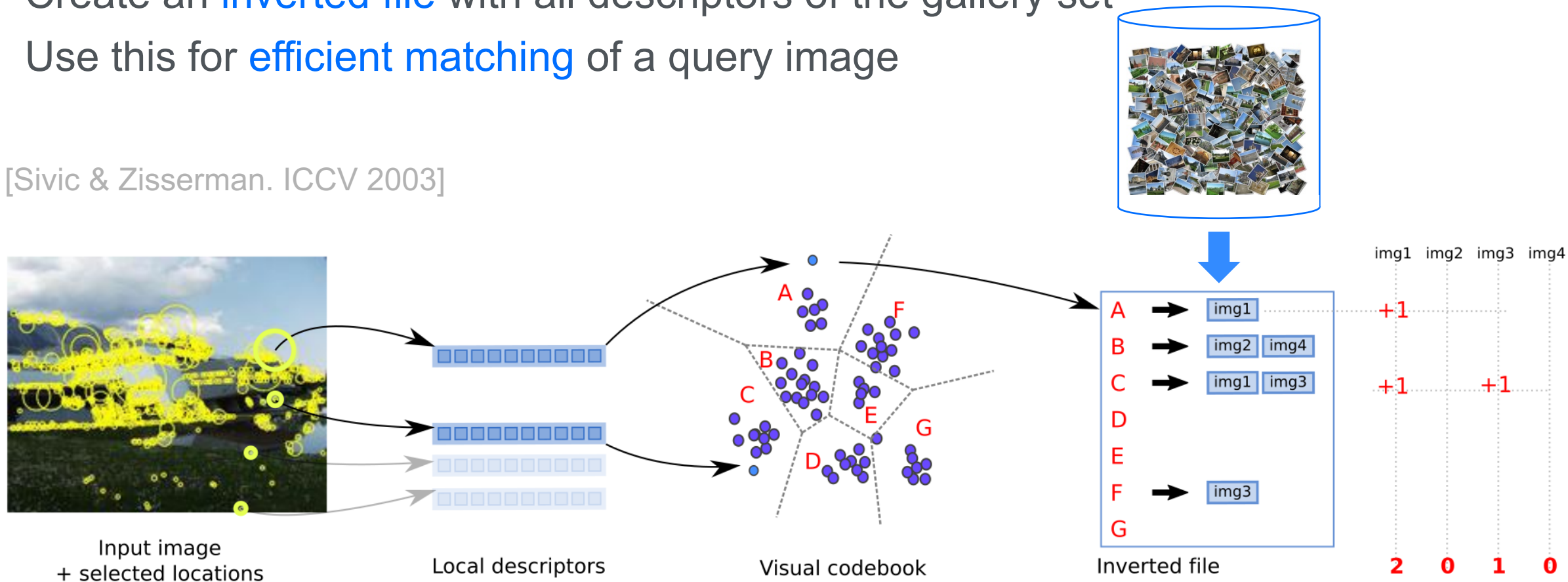


Leverage quantization for efficient retrieval

Create an **inverted file** with all descriptors of the gallery set

Use this for **efficient matching** of a query image

[Sivic & Zisserman. ICCV 2003]



Evaluation on standard benchmarks

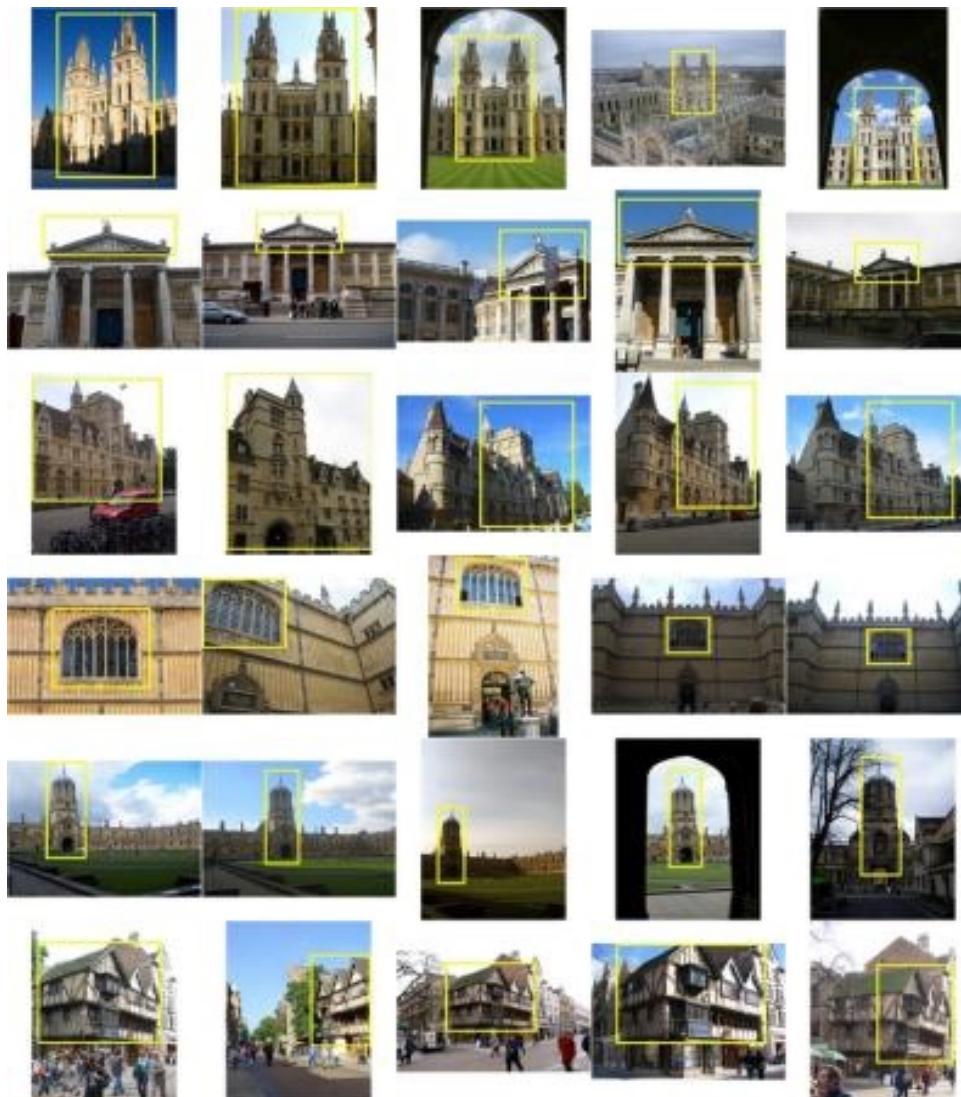
Oxford dataset

- 5,000 images
- 55 queries
- 11 landmarks

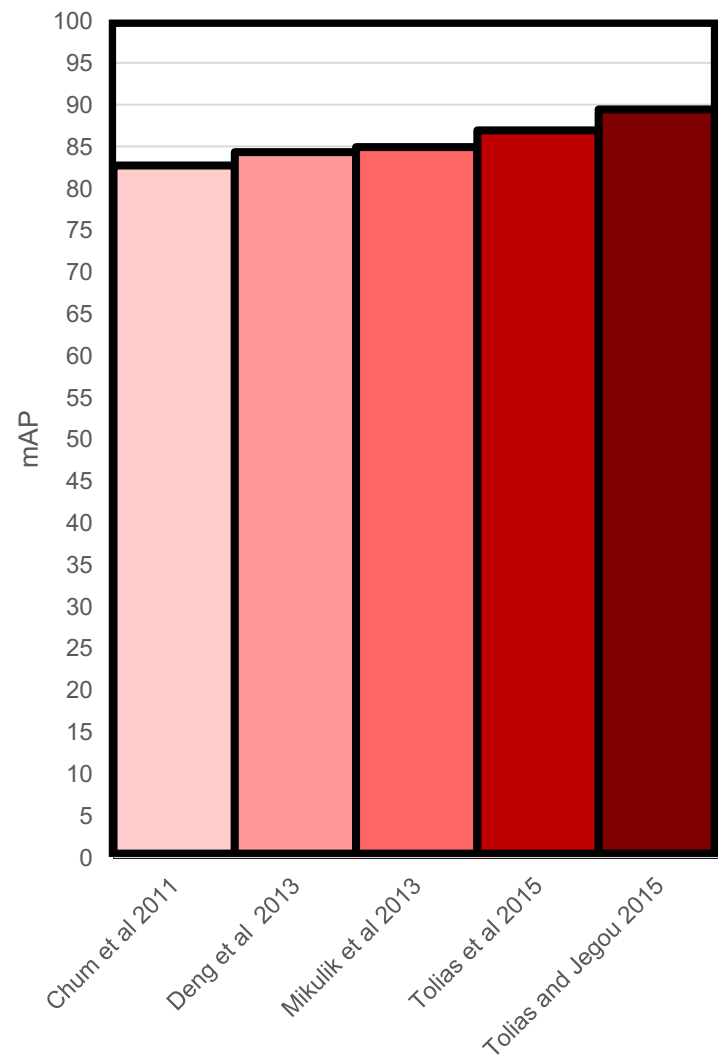
Evaluation

- mean Average Precision (mAP)

[Philbin et al. CVPR07]



Experiments



Oxford 5K



Local representations

Description locale des images : résumé et commentaires

- Étapes
 - ▶ recherche de points/régions d'intérêt
 - ▶ calcul des descripteurs pour chacune de ces régions
 - ▶ vérification de la cohérence géométrique
 - ▶ algorithme de vote
- Commentaires:
 - ▶ problème : il faut apparier les descripteurs d'une image avec tous les descripteurs d'une base d'image
→ c'est très lent !
 - ▶ la vérification de la cohérence géométrique est relativement lente
→ on ne peut l'utiliser que pour un sous-ensemble d'images *a priori* plus pertinentes

Limitations

This full pipeline is a costly approach

- **Stores all the local descriptors** – significant memory, even after compression
- **Matching and verification** represent a significant computational cost

Alternative approach

- **Global descriptors** aggregate individual local descriptors per image
- Simple similarity measure between global descriptors

Merci