Comprendre les données visuelles à grande échelle

ENSIMAG
2019-2020

Karteek Alahari & Diane Larlus

17 octobre 2019

Description locale des images

Comprendre les données visuelles à grande échelle

Cours 1: Introduction, 17 octobre 2019

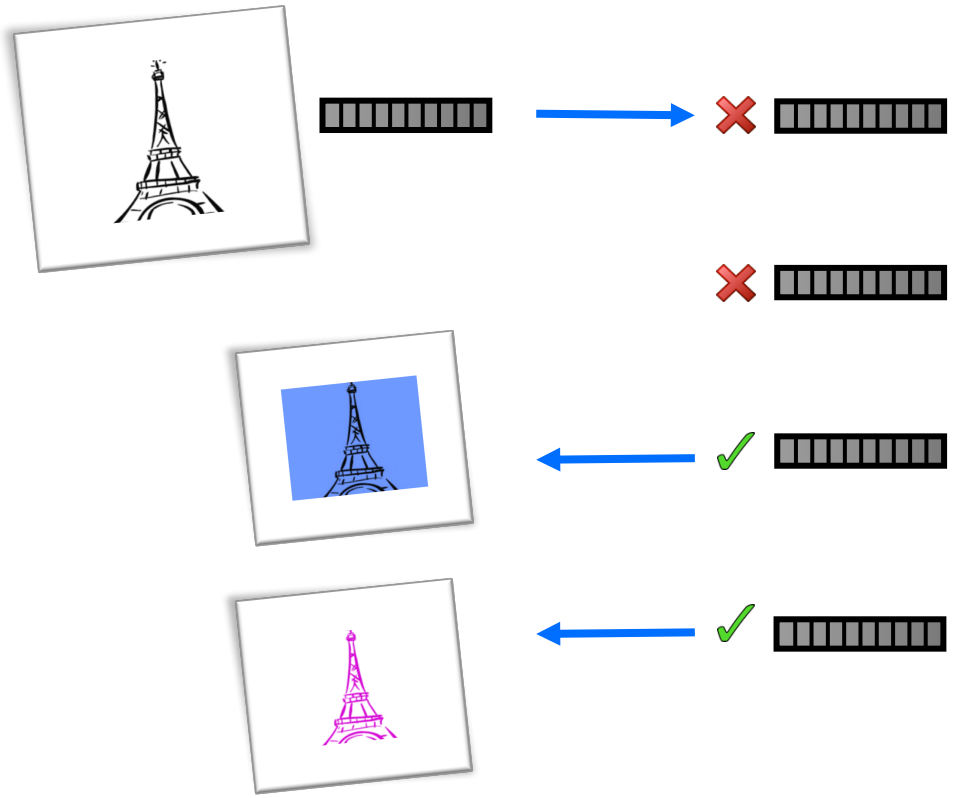
NAVER LABS EUROPE

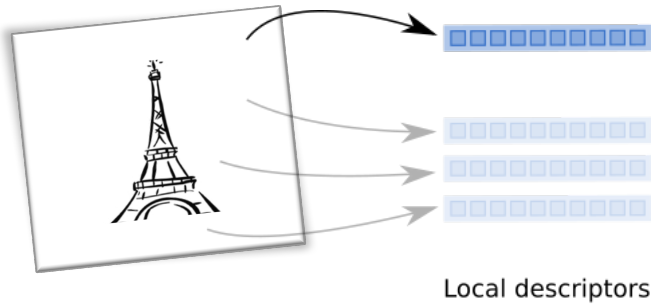
Rappel: variation d'apparence d'une **instance** d'objet donné

NAVER LABS EUROPE

Local representations

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





NAVER LABS EUROPE

Local representations

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

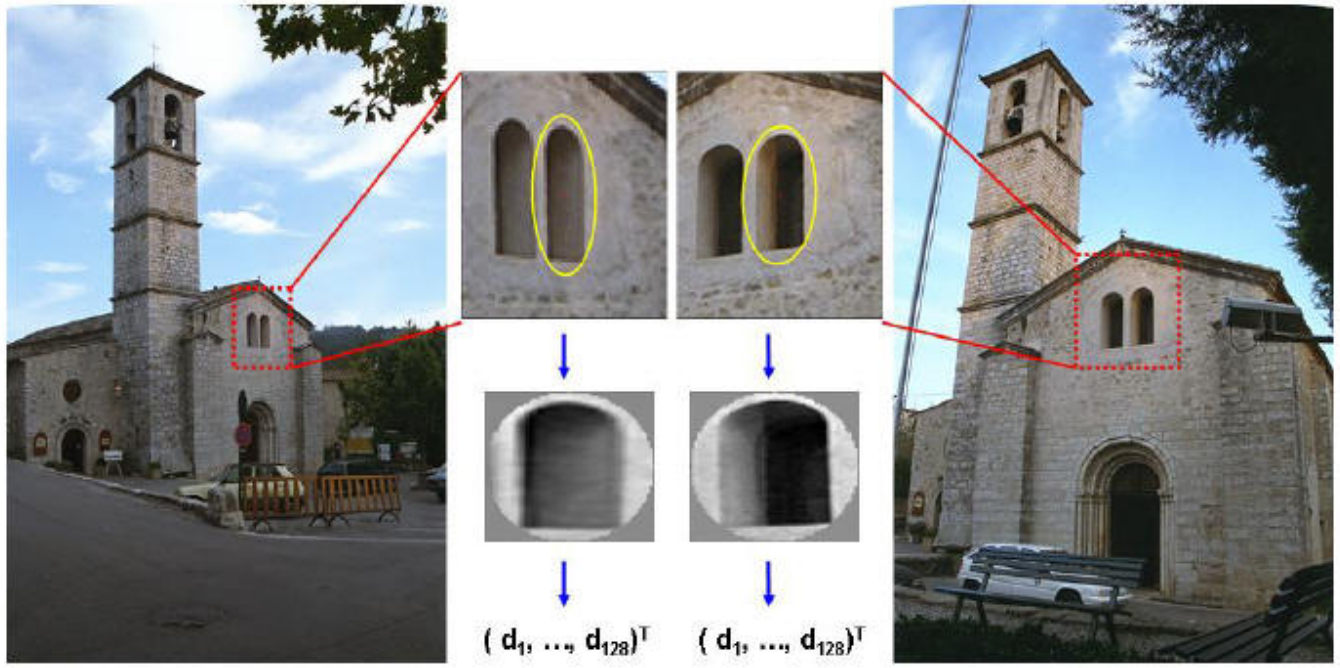
Local representations seem more adapted but require more sophisticated matching

Représentations locales

- Les représentations locales captures l'apparence de l'image localement
- L'apparence locale est extraite à plusieurs positions dans l'image, choisies avec soin, notamment à l'aide d'un critère de « **répétabilité** ».

Définition: un détecteur est dit répétable s'il choisit les mêmes régions d'une scène / d'un objet dans des images potentiellement très différentes de cet objet

Exemple: la fenêtre est choisit par le détecteur dans les deux images



NAVER LABS EUROPE

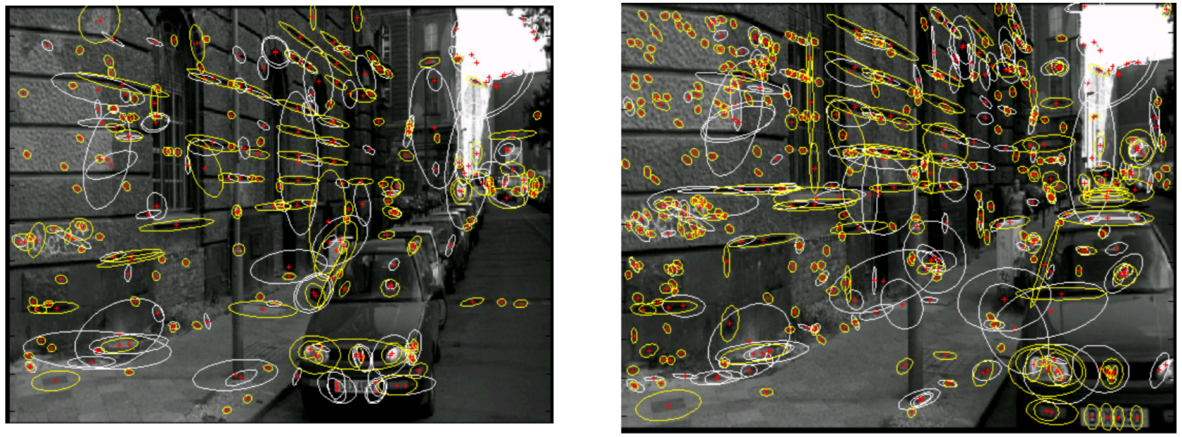
Représentations locales

- Les représentations locales captures l'apparence de l'image localement
- L'apparence locale est extraite à plusieurs positions dans l'image, choisies avec soin, notamment à l'aide d'un critère de « **répétabilité** ».

Définition: un détecteur est dit répétable s'il choisit les mêmes régions d'une scène / d'un objet dans des images potentiellement très différentes de cet objet

On utilise pour cela des détecteurs de points d'intérêt

- Exemple de détecteurs: Harris, Hessian, Hessian-Affine, MSER, etc.

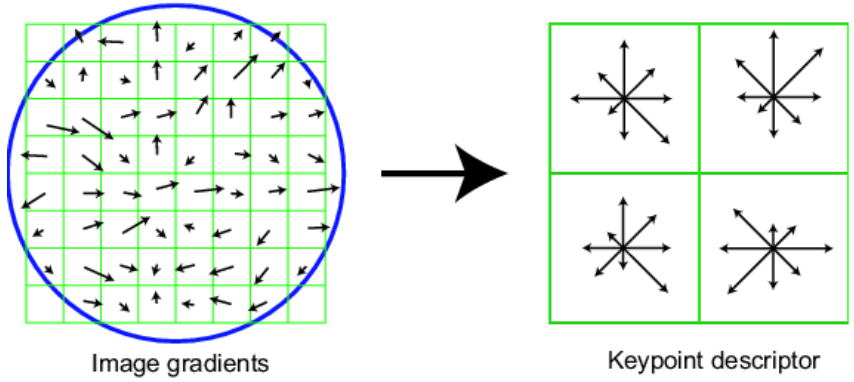


[Harris&Stephens1988, Mikolajczyk&Schmid2002, Matas et al, 2004]
Survey: [Mikolajczyk, IJCV 2005]

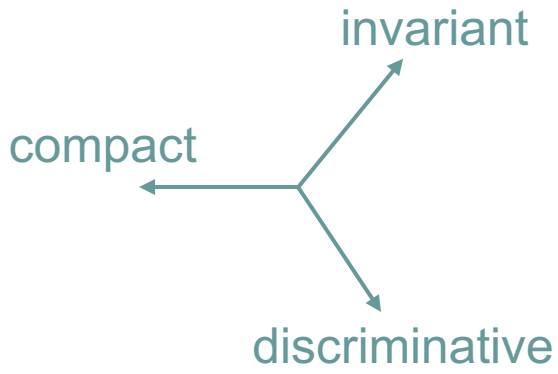
Représentations locales

- Les représentations locales captures l'apparence de l'image localement
- Ces représentations locales doivent posséder des propriétés d'invariance et de discrimination

Local descriptors
- SIFT, SURF, LBP, etc.



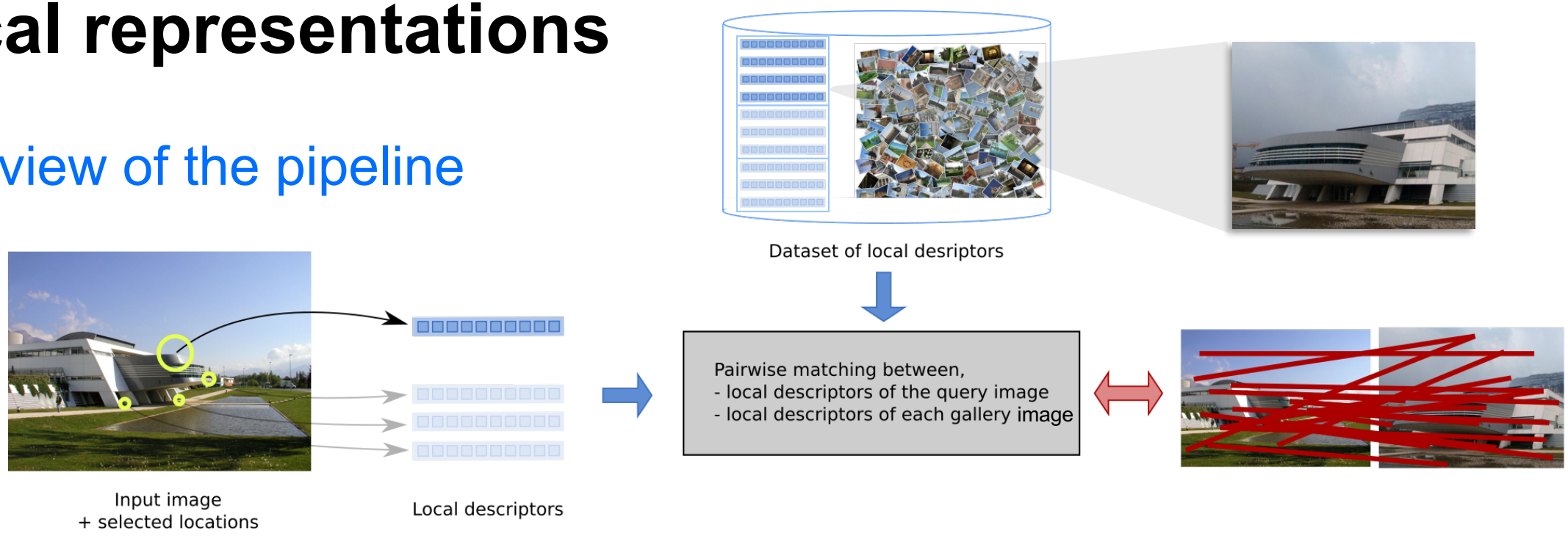
[Lowe IJCV04, Bay et al, 2008, Ojala et al, 2014]



(plus de détails à venir)

Local representations

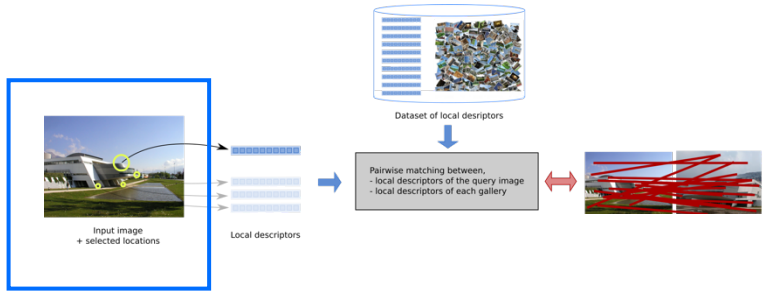
Overview of the pipeline



NAVER LABS EUROPE

Description locale - plan

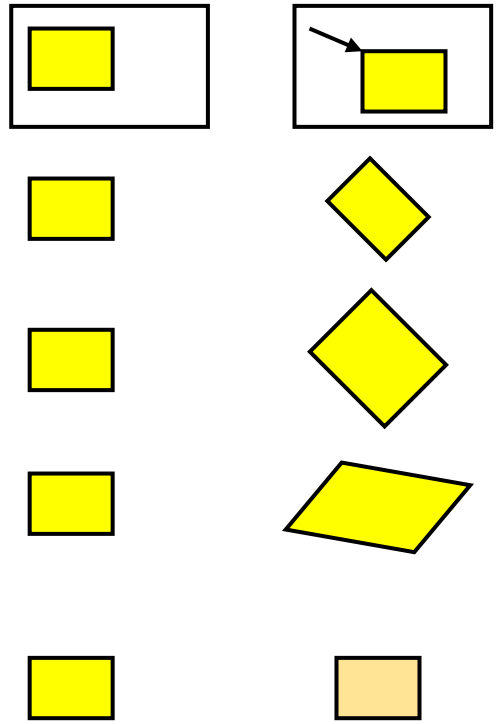
- Extraction de points/régions d’intérêt
- Descripteurs locaux
- Appariement de points



Extraction de points/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 planaire)



- Transformations photométriques
  - Changement d'intensité affine (I –> a I + b)

Principaux extracteurs de points/régions d'intérêt

Vus en cours

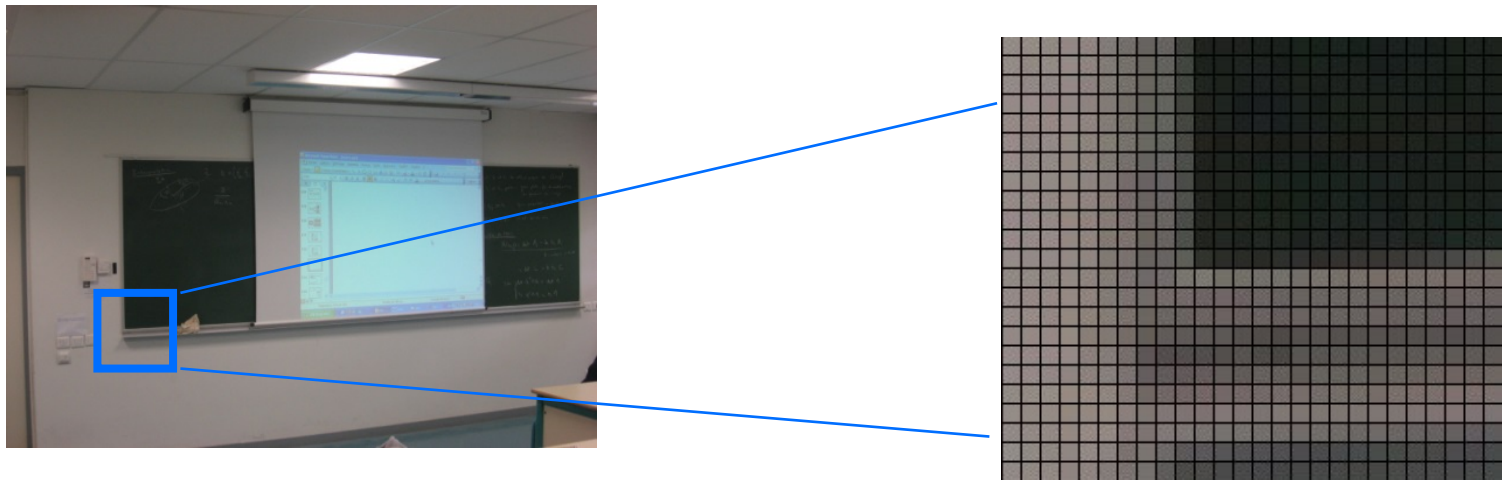
- *Harris*
- *Harris-Affine*
- *Hessian-Affine*
- *Maximally Stable Extremal Regions (MSER)*
- *Salient Region Detector*

[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

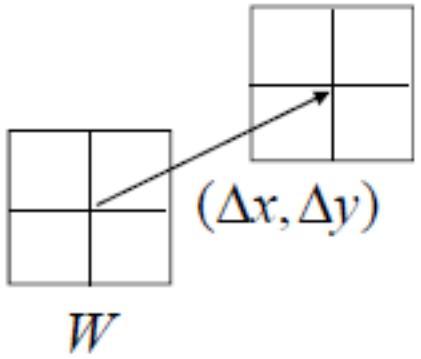


NAVER LABS EUROPE

Détecteur de Harris

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

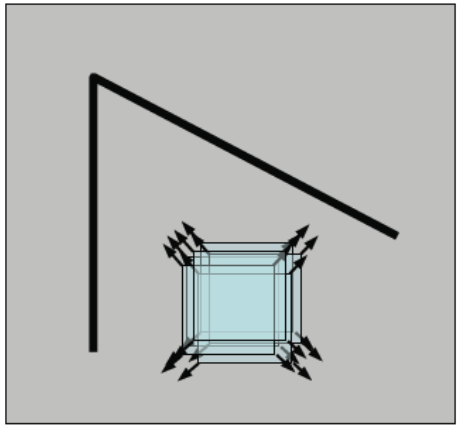
$$A(x, y) = \sum_{(x_k, y_k) \in W(x,y)} (I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y))^2$$



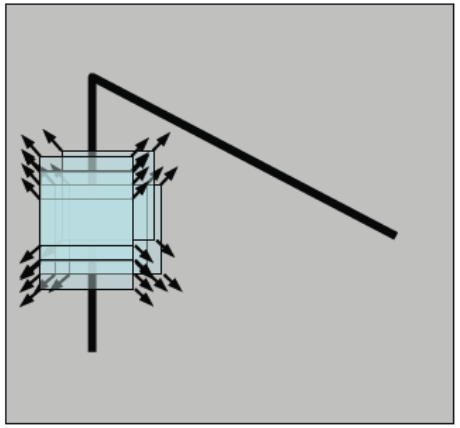
- *A(x,y)*:
  - Faible dans toutes les dimensions -> région uniforme
  - Large dans une direction -> contour
  - Large dans toutes les dimensions -> point d'intérêt

Détecteur de Harris

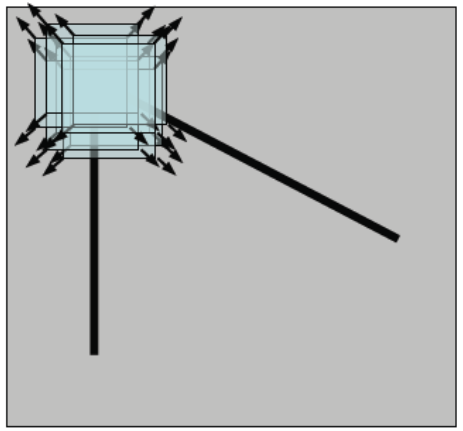
- Illustration:



"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 ?
- Basée sur une **approximation du premier ordre**

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

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

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

Détecteur de Harris

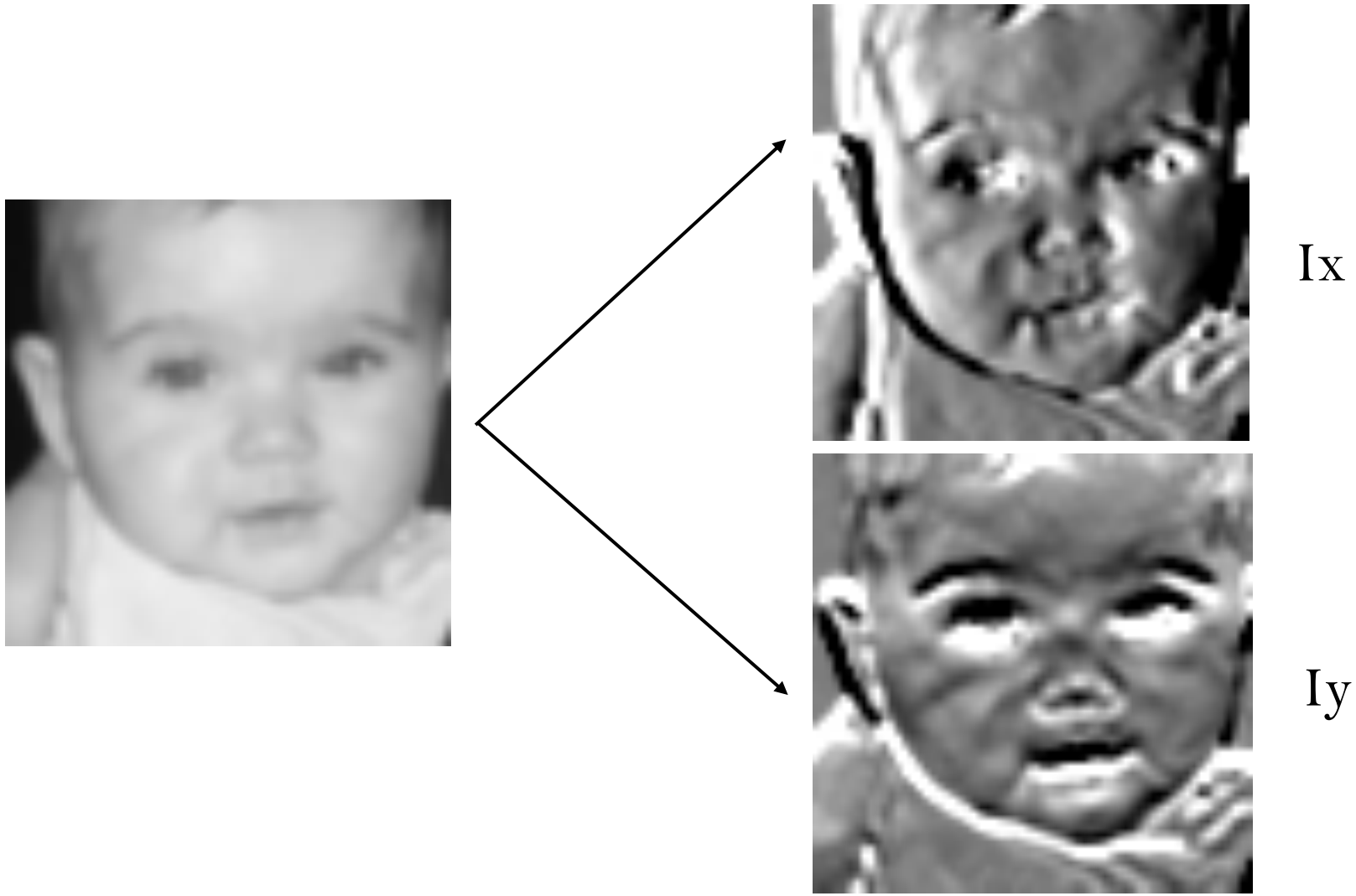
- Basé sur une approximation du premier ordre

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

Auto-correlation matrix

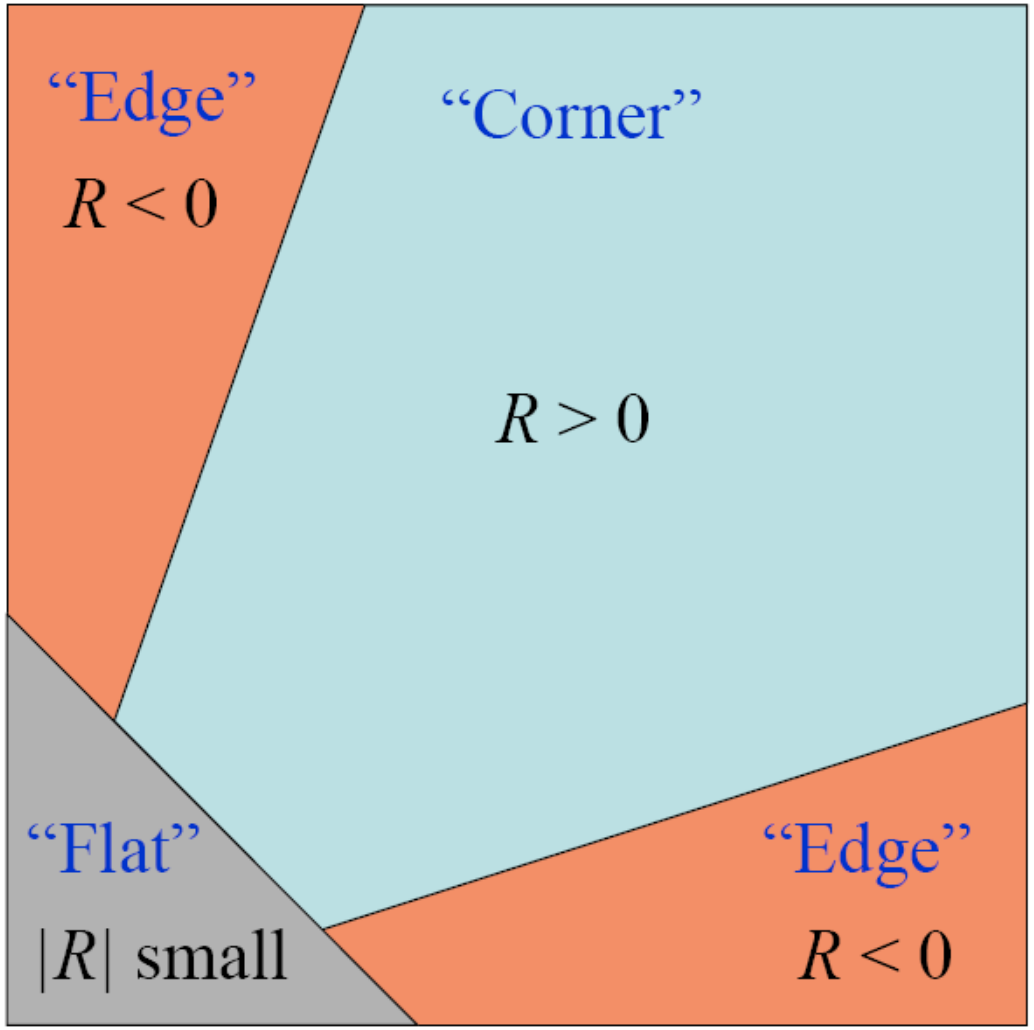
Détecteur de Harris : calcul des gradients

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



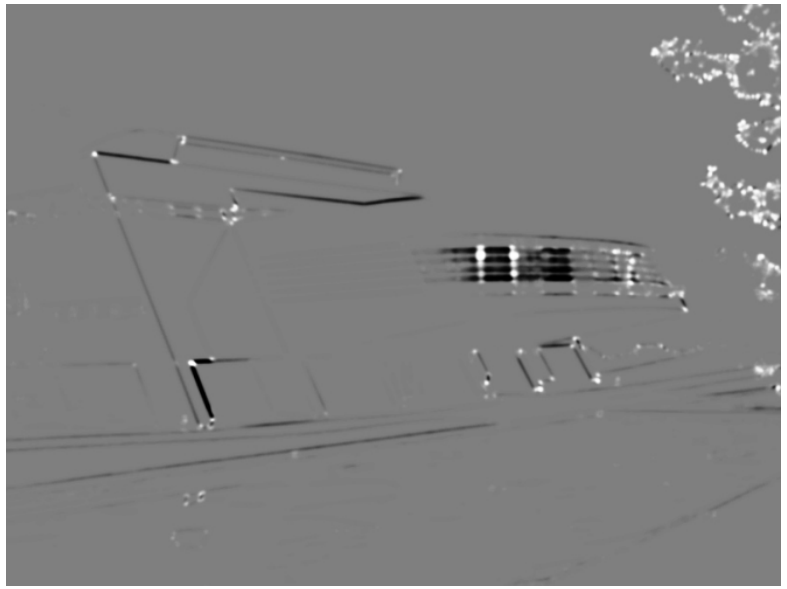
Détecteur de Harris : classification des régions par utilisation d'un seuil

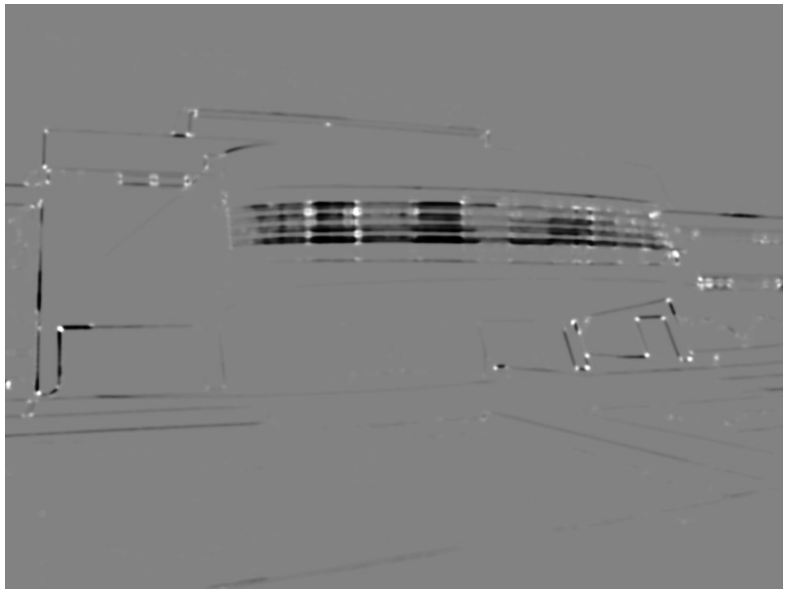
$$R = \det(A) - \alpha \,\mathrm{trace}(A)^2 = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$



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

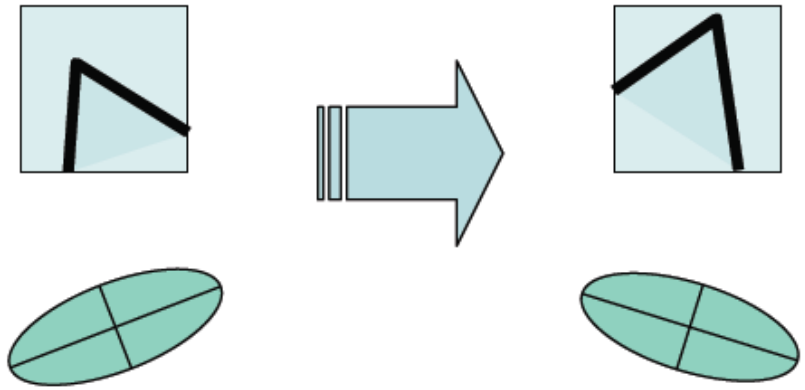
Détecteur de Harris : exemple





Détecteur de Harris : propriété d'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

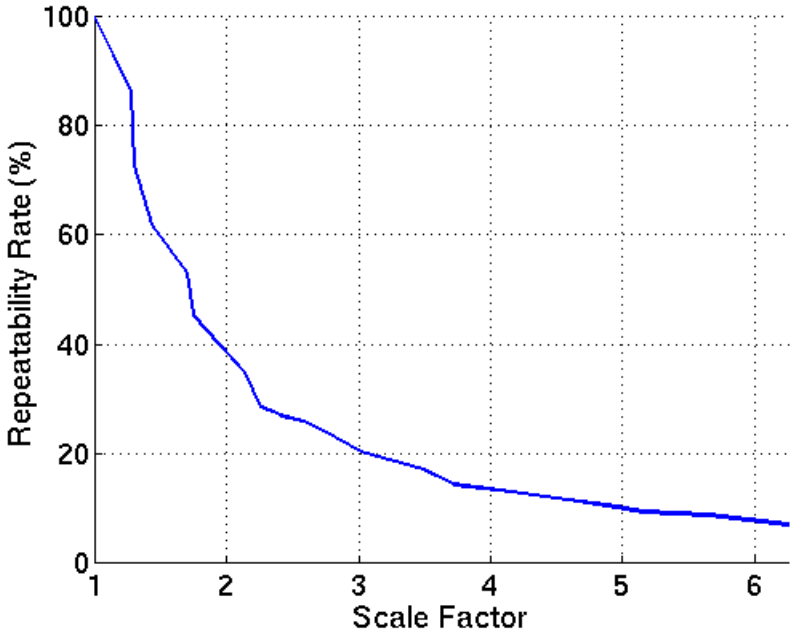


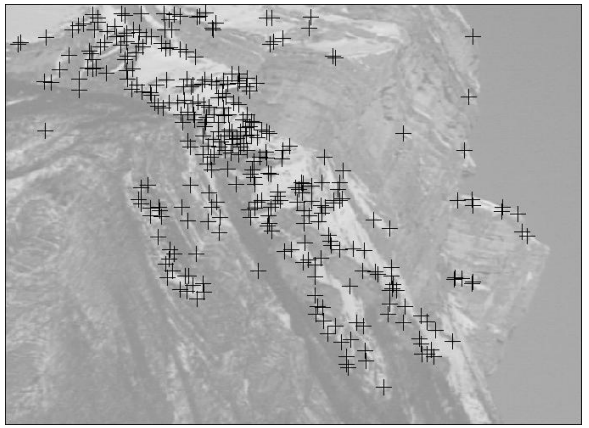
- **Mais pas d'invariance à des changements d'échelle**

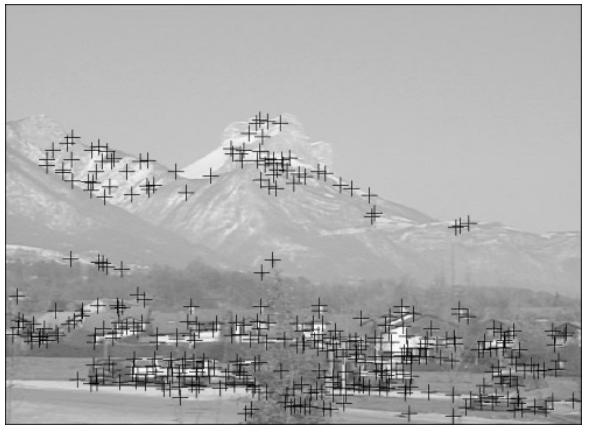
Détecteur de Harris : Invariance à l'échelle

- Pas d'invariance aux changements d'échelle

- Taux de répétabilité



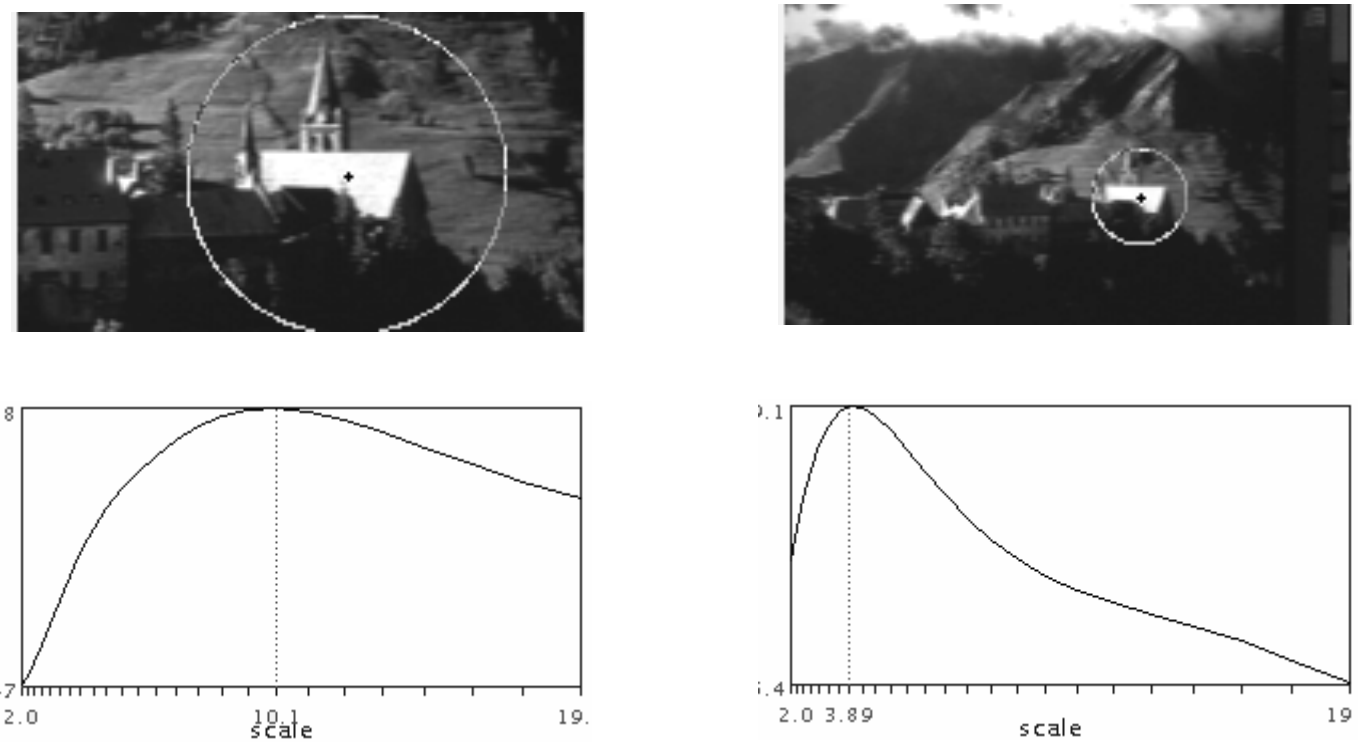




Détecteur de Harris : Invariance à l'échelle

- Choix d'un 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

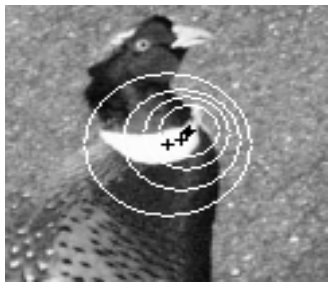
$$\nabla \cdot \nabla f = \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$



invariance de l'échelle caractéristique

Détecteur de Harris : Invariance à l'échelle

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

  - Sélection des points à leur échelle caractéristique avec le Laplacien


Détecteur de Harris-Affine (début)

- **Détecteur de région de l'état de l'art**

- Algorithme : estimation itérative des paramètres

  - Localisation du point d'intérêt: utilisation du détecteur de Harris
  - Choix de l'échelle : sélection automatique avec le Laplacien
  - **Choix d'un voisinage affine :**
    normalisation avec la matrice des seconds moments

  –> jusqu'à convergence

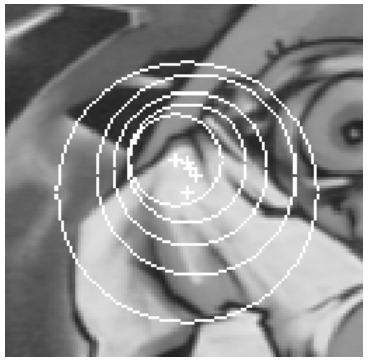
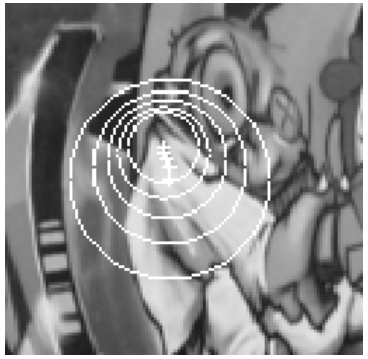
Détecteur de Harris-Affine (suite)

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

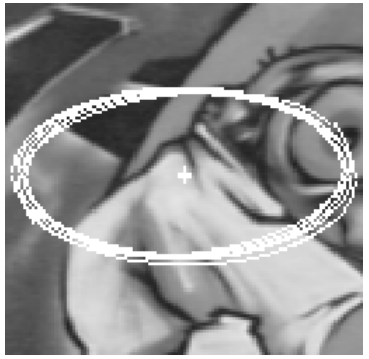
Iteration #3, #4, ...

Détecteur de Harris-Affine: pour résumer

- Initialisation avec des points d'intérêt multi-échelle grâce au détecteur de Harris

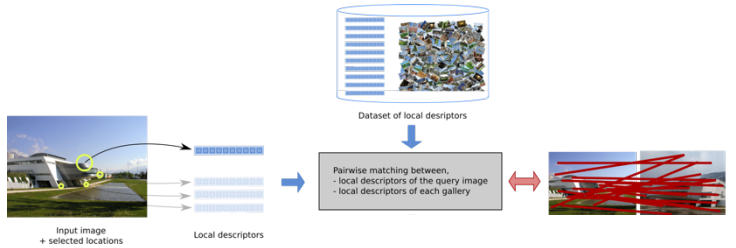



- Estimation itérative de la localisation, de l'échelle, du voisinage

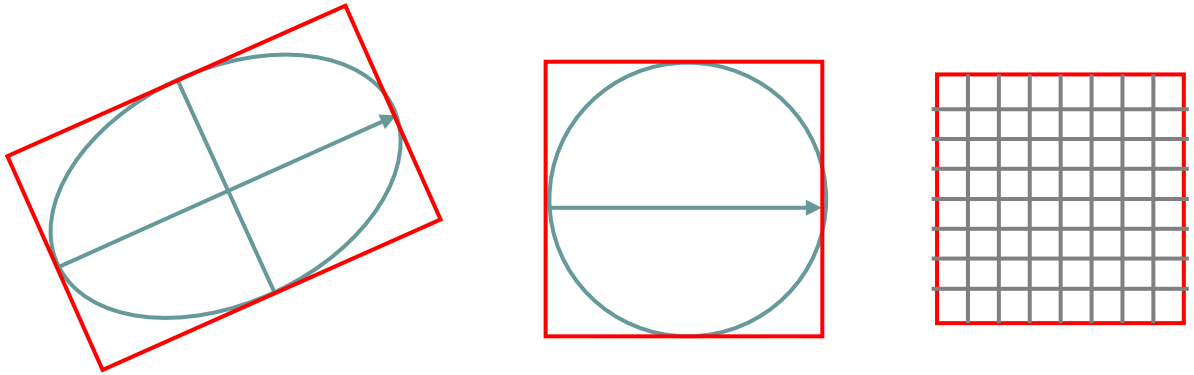
Description locale : plan

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

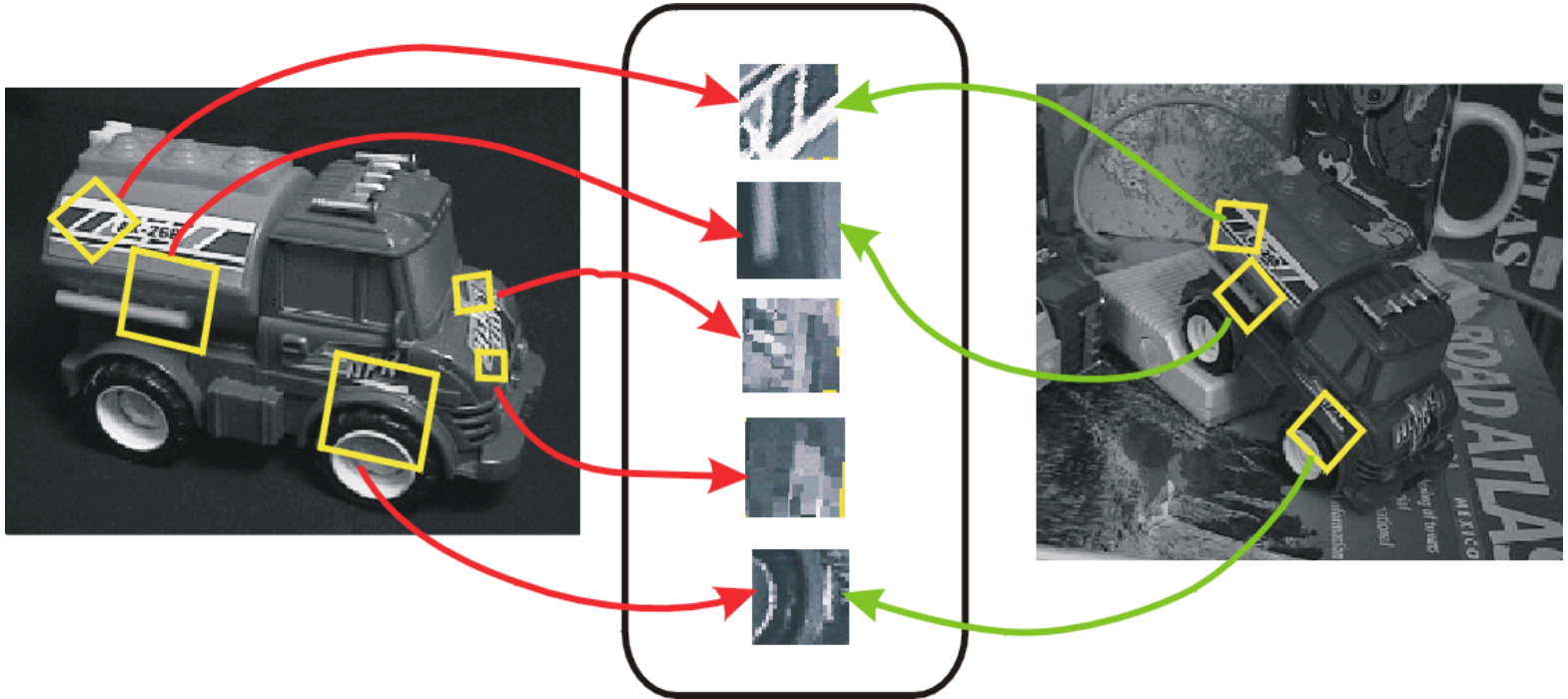


Et une fois qu'on a défini l'ellipse ?

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



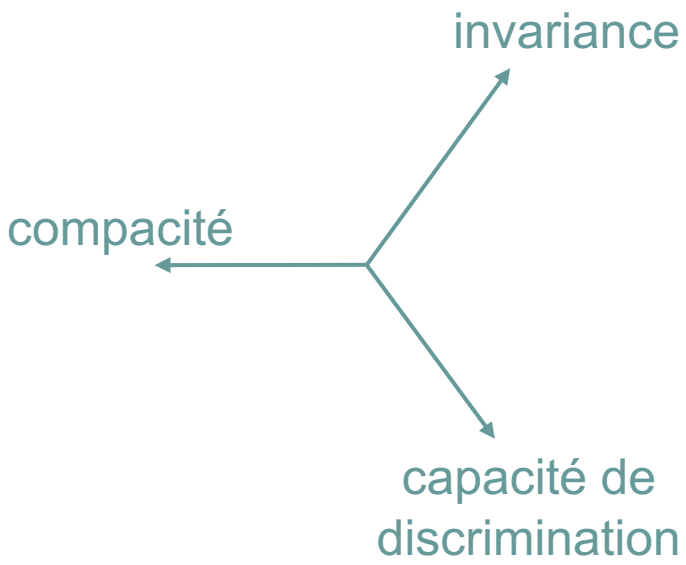
- Tailles typiques de patch : 21x21, 64x64



NAVER LABS EUROPE

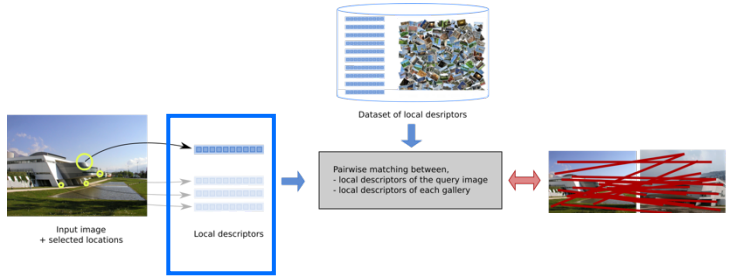
Objectif

- 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



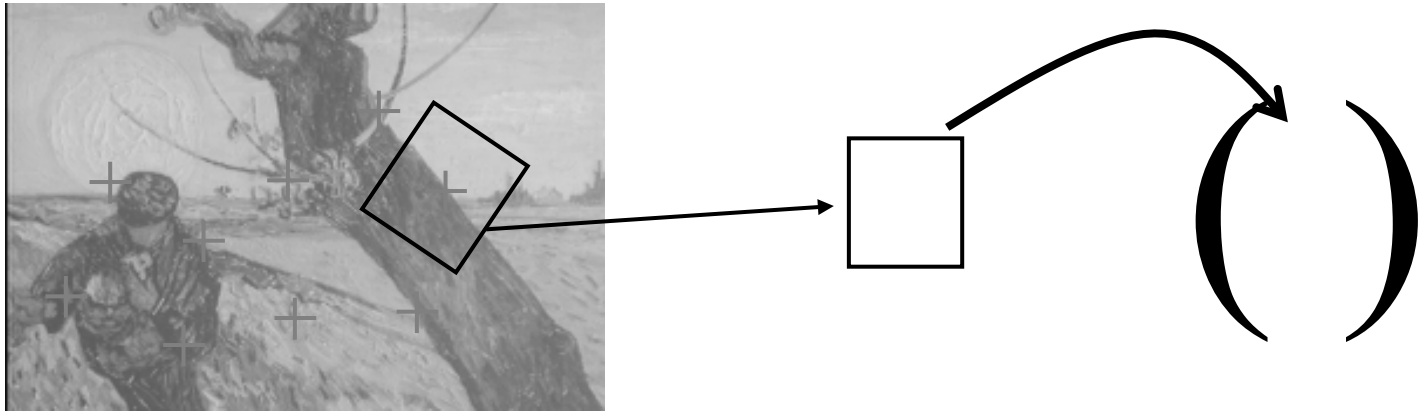
Description locale : plan

- Extraction de points/régions d'intérêt
- Descripteurs locaux
  - Descripteurs basiques
  - État de l'art : le descripteur SIFT
  - Descripteur CS-LBP
- Appariement de points



Descripteurs de pixels

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



descripteur local

- Calcul simple
- Volumineux (patch de 32x32 = vecteur de 1024 éléments)
- Invariance possible aux transformations affines de luminosité
- Mauvaise « baseline »

SIFT (Scale invariant feature transform)

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

(explications au tableau)

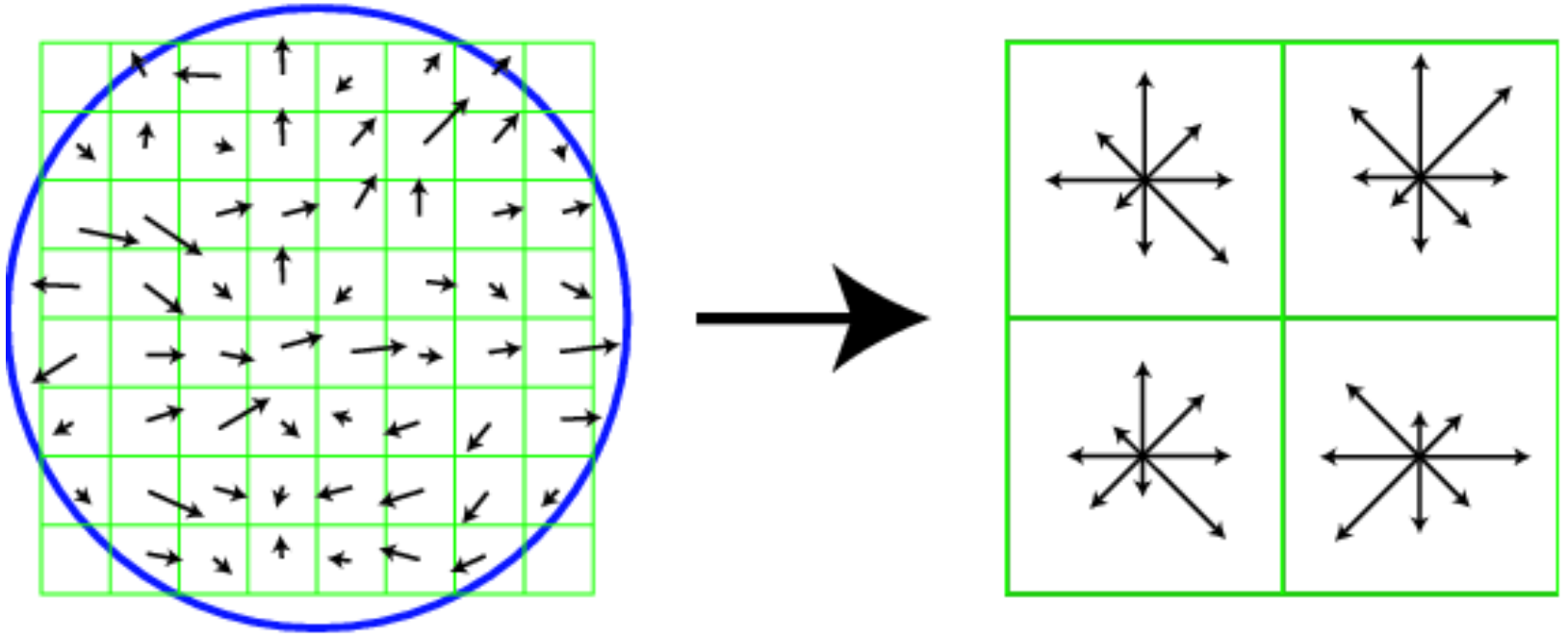
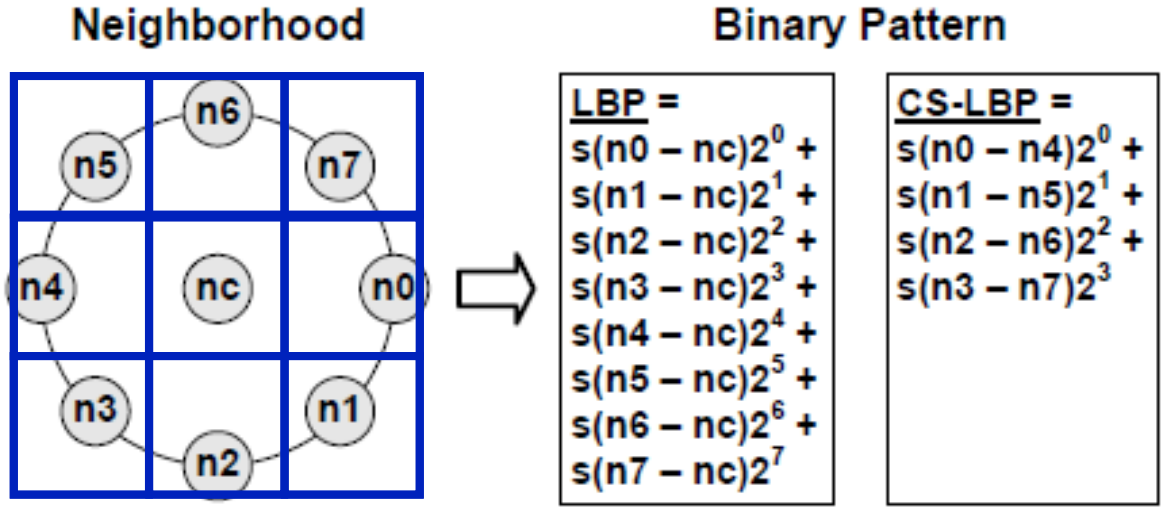


Image gradients

Keypoint descriptor

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

- Principe du descripteur CS-LBP
  - Le patch est divisé en une grille de cellules
  - Pour chaque pixel dans la grille
    - Le comparer à ses 8 voisins
    - Encoder ces 8 comparaisons (0 ou 1) en un nombre binaires à 8 chiffres
  - Pour chaque cellule
    - Construire un histogramme d'occurrence de ces nombres binaires
    - Cela donne un histogramme à 256 dimensions
  - Concaténer les histogrammes de chaque cellule



- Il existe de 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

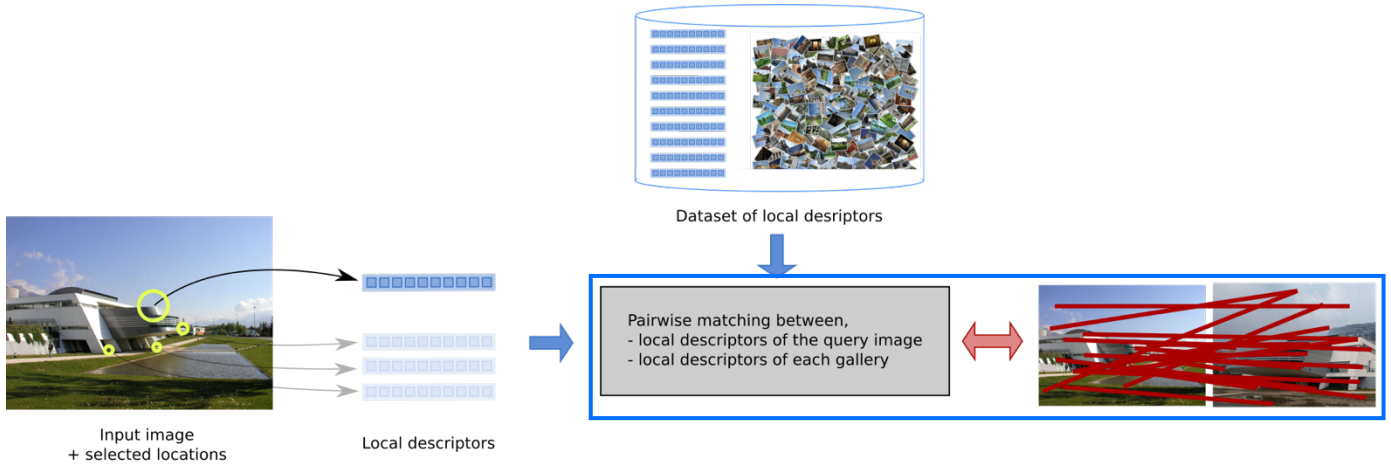
Descripteurs locaux: conclusion

- Choisir un détecteur avec de nombreuses **invariances** par construction
  - 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 (par exemple 128 pour SIFT) pour obtenir une bonne capacité de **discrimination**

Description locale : plan

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



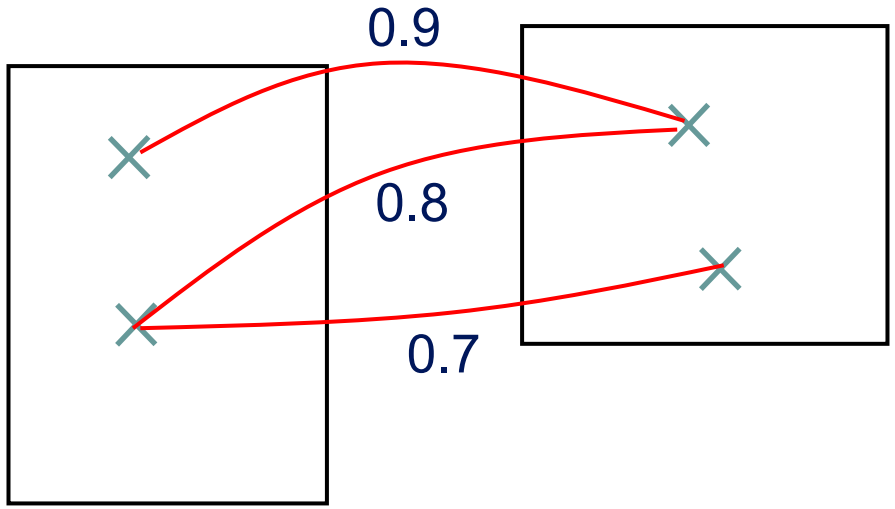
Appariement de points

- 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 le suite: sélection des meilleurs couples de points

Sélection des meilleurs couples

- Remarque: ambiguïté

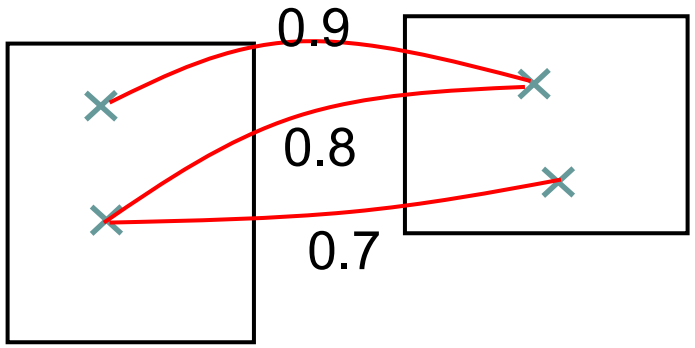


- Différentes stratégies:
  - *Winner-takes-all* (symétrique ou à sens unique)
  - Filtrage ultérieur par géométrie

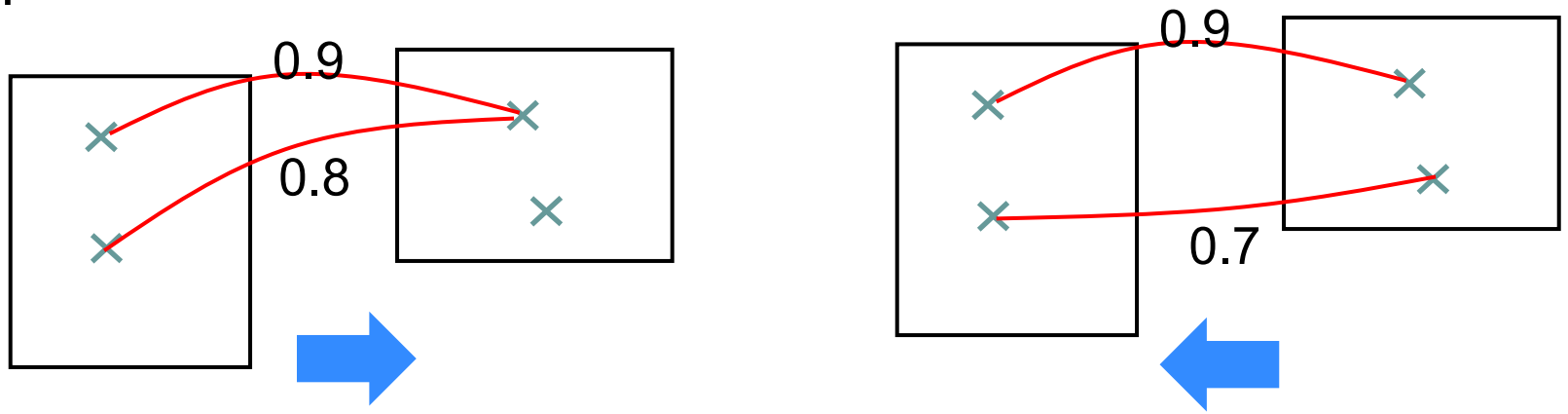
Sélection des meilleurs couples

Différentes stratégies:

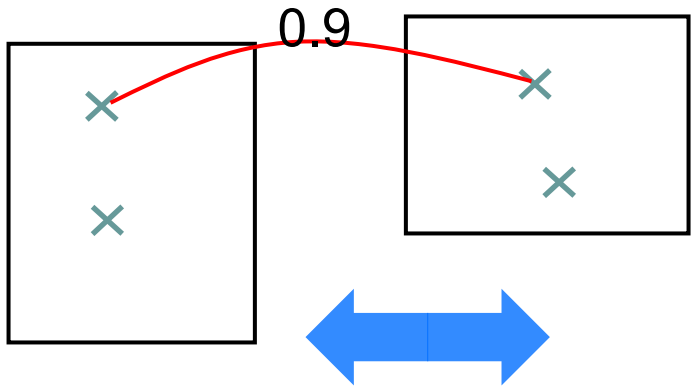
- ***Winner-takes-all***



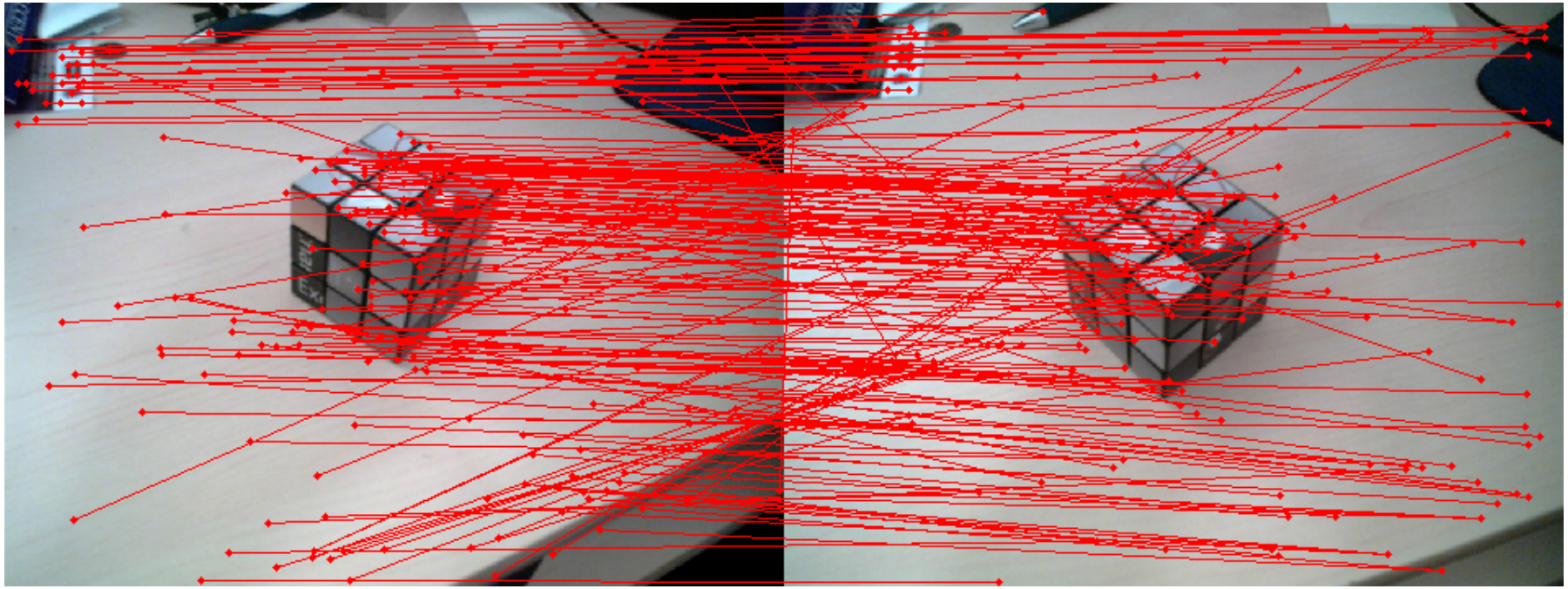
A sens unique



Symétrique



Problème : appariements entre deux images



NAVER LABS EUROPE

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

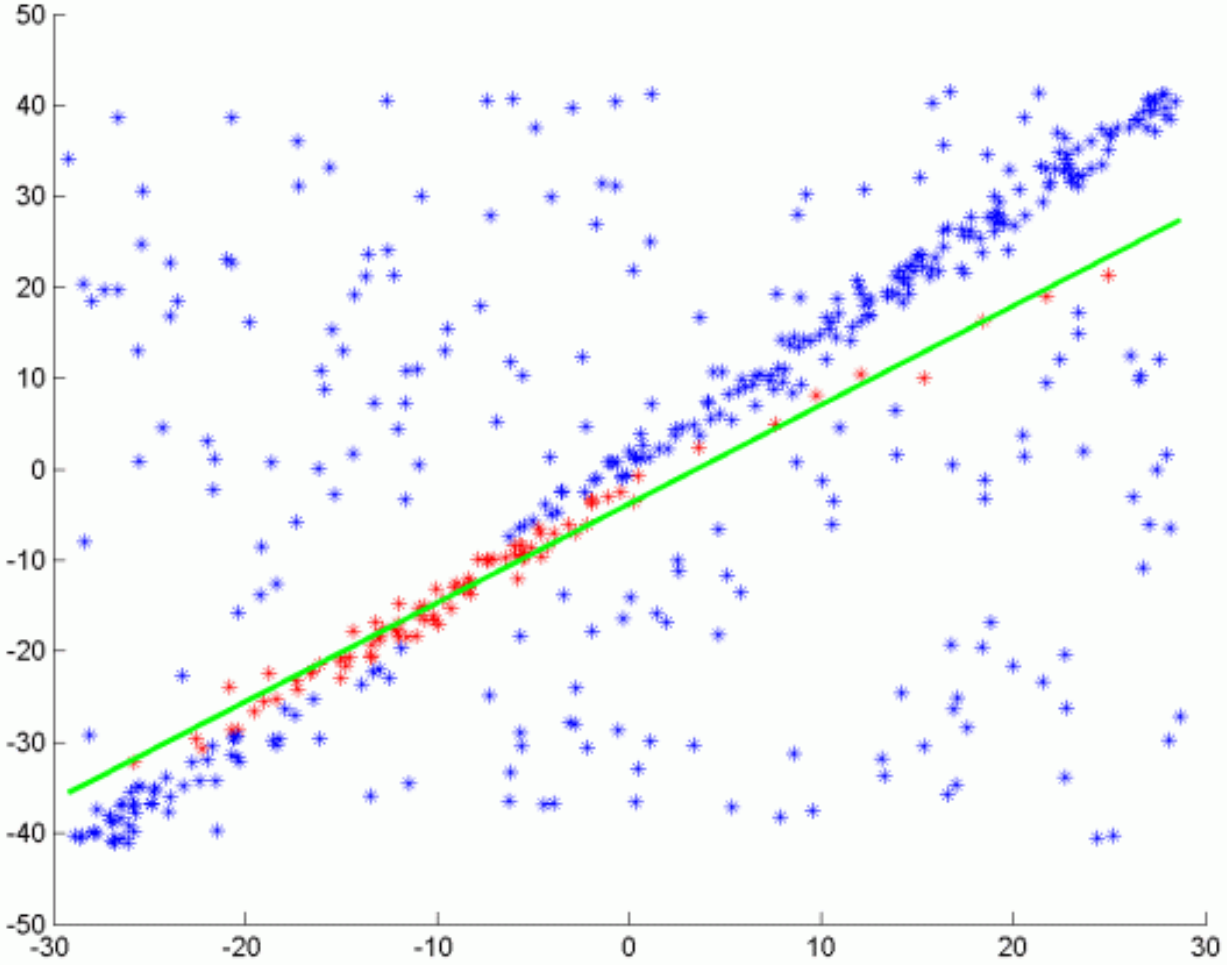
RANSAC

RANSAC (*RANdom SAmple Consensus*)

- 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*

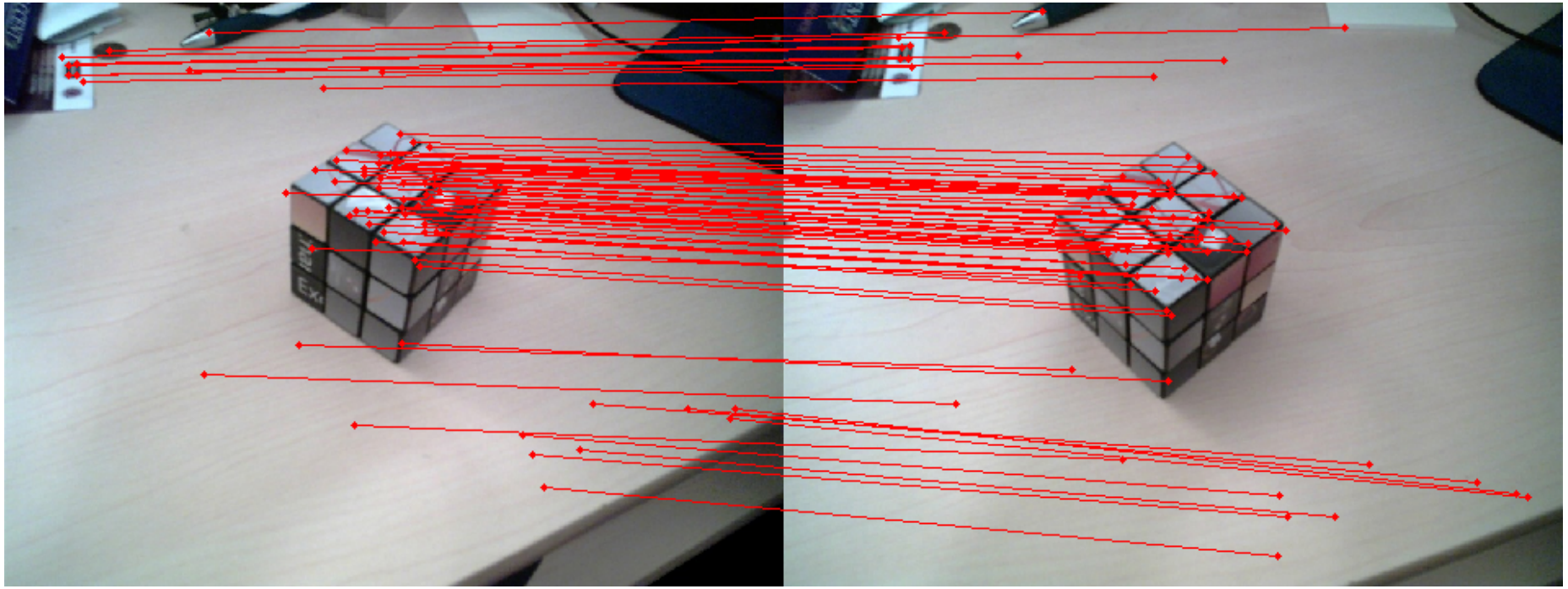
RANSAC

RANSAC (*RANdom SAmple Consensus*)



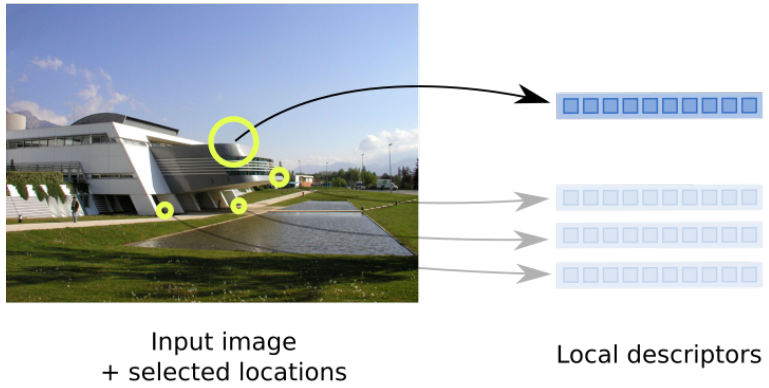
Problème : appariements entre deux images

après RANSAC



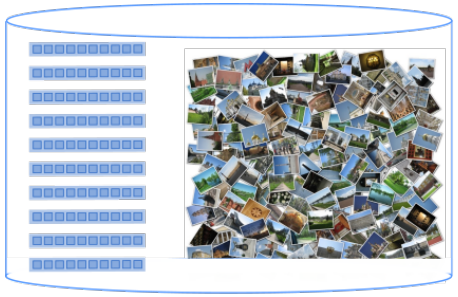
Local representations

Summary of the pipeline



Input image
+ selected locations

Local descriptors



Dataset of local desriptors

Bottleneck: pairwise matching

Pairwise matching between,
- local descriptors of the query image
- local descriptors of each gallery

Short list of images with enough pairwise matches

Geometric verification (RANSAC) of the short list of images

Shorter list of retrieved images

NAVER LABS EUROPE

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

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

- Locality-Sensitive Hashing (LSH)
  - Randomized hashing technique
  - Data-dependent variants: spectral hashing, semantic hashing

[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

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

- Locality-Sensitive Hashing (LSH)
  - Randomized hashing technique
  - Data-dependent variants: spectral hashing, semantic hashing

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

2) Quantize local descriptor space into a visual codebook

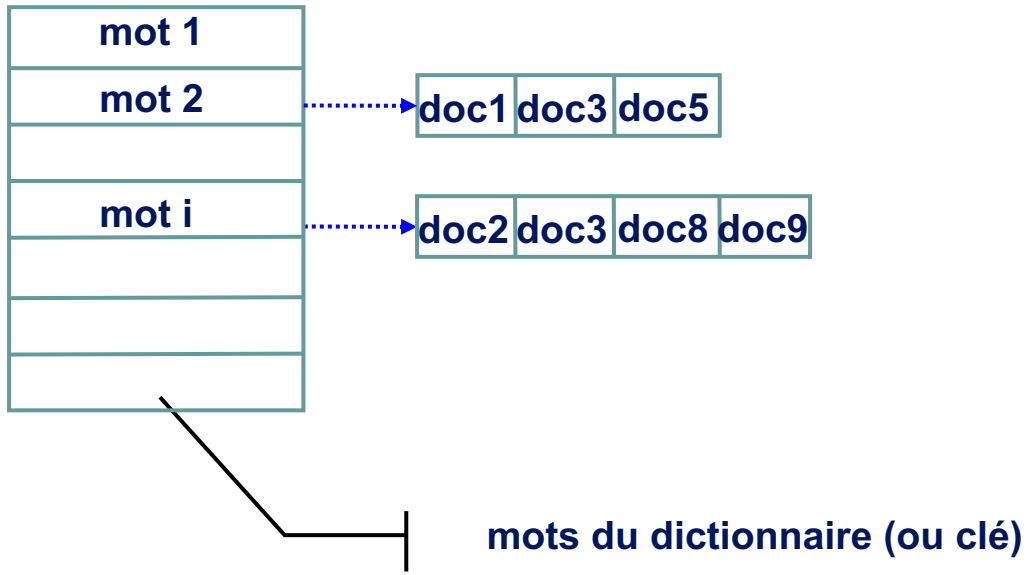
- Uses **inverted files** to store the descriptors
- Efficient image matching

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 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
- 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
    $f = (f_1,\ldots,f_i,\ldots f_d) \in \mathbb{R}^d$
  - 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)

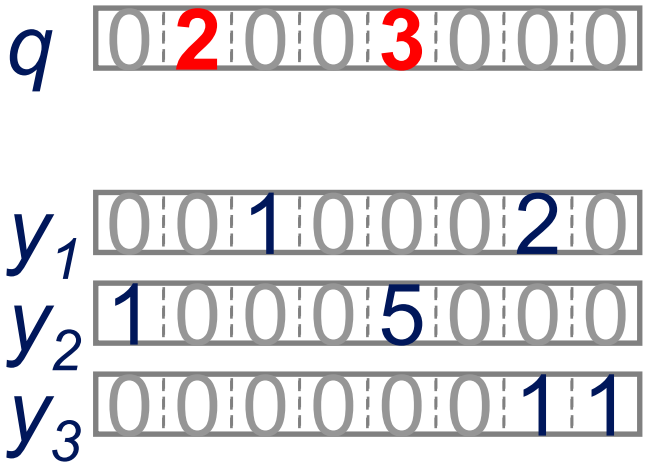
- Exemple
  - vocabulaire = {“vélo”, “voiture”, “déplace”, “travail”, “école”, “Grenoble”}
  - espace de représentation: $\mathbb{R}^6$
  - “Grenoble est une belle ville. Je me déplace à vélo dans Grenoble”

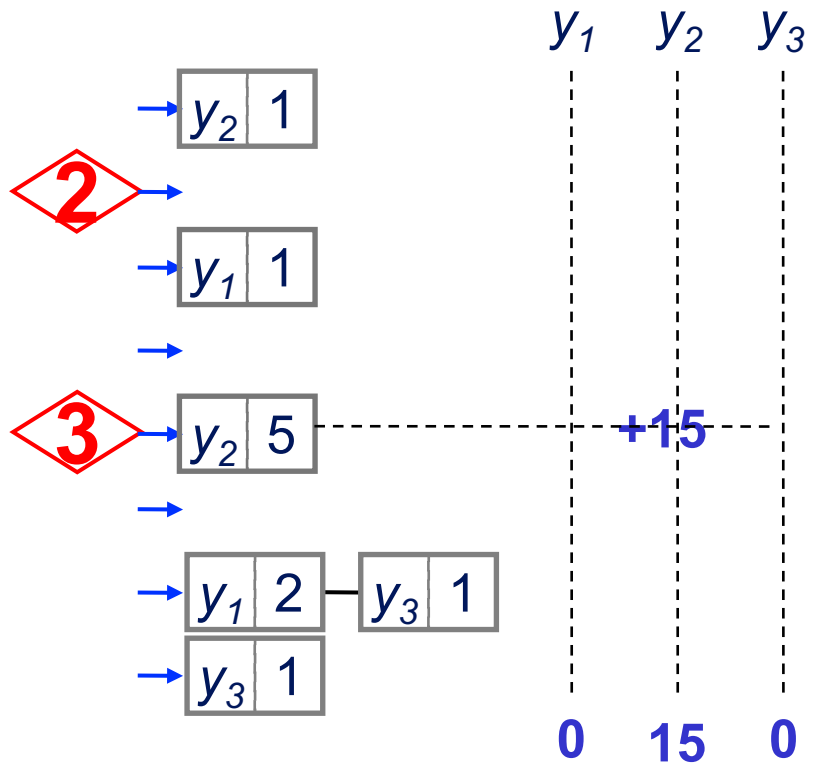
  $f=(1/4,0,1/4,0,0,1/2)^t$

- Recherche de document = trouver les vecteurs les plus similaires à un document requête, représenté par le vecteur *q*
  - 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





NAVER LABS EUROPE

Recherche de documents visuels

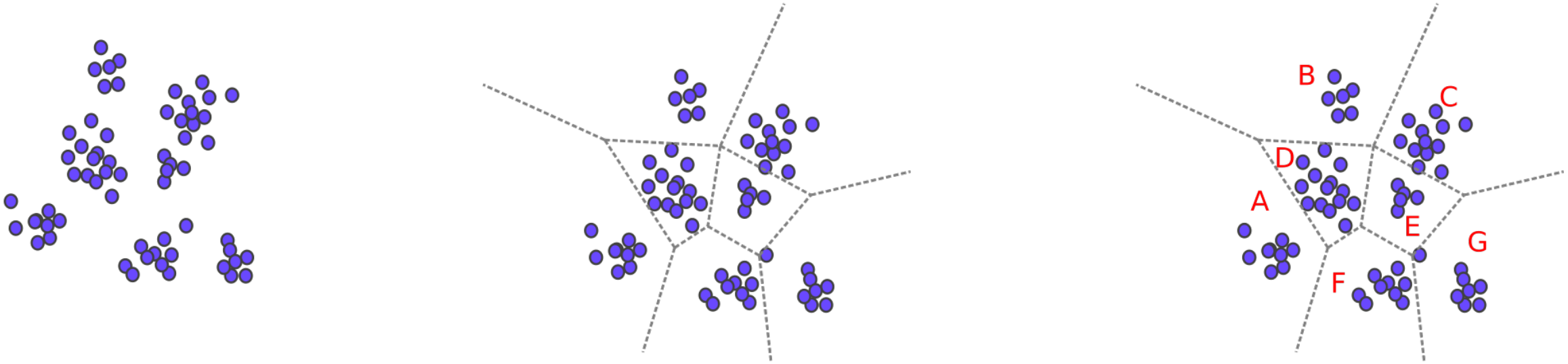
- Problème: l'espace des descripteurs locaux est continu
  (par exemple l'espace des SIFT)
  - Il existe potentiellement une infinité de descripteurs locaux possibles
  - Il n'existe pas de concept de vocabulaire = 'ensemble de mots' pour les *patchs* d'images tel que c'est le cas pour le texte

- Solution: créer un **vocabulaire visuel** !
  - Aussi appelé *visual vocabulary* or *visual codebook* dans la littérature
  - Construction par quantification de l'espace des descripteurs locaux

Quantization: principle

Principle:

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



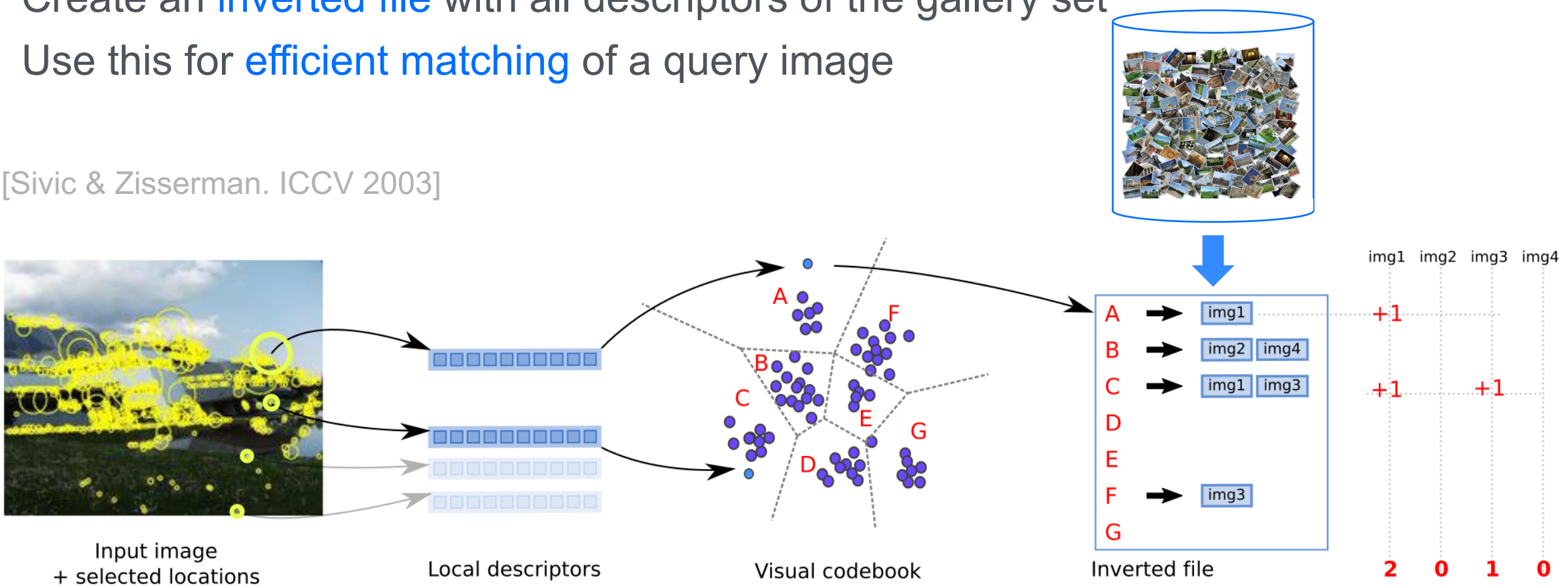
NAVER LABS EUROPE

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]



NAVER LABS EUROPE

Description locale des images : résumé et commentaires

- **Les différentes étapes**
  - Sélection de points ou de régions d'intérêt (ex: Harris ou Harris-Affine)
  - Calcul des descripteurs pour chacune de ces régions (ex: SIFT, CS-LBP)
  - Algorithme de mise en correspondance entre images (ex: calcul exhaustif des distances entre paires de descripteurs, utilisation d'un fichier inversé)
  - Vérification de la cohérence géométrique (ex: RANSAC)

Description locale des images : résumé et commentaires

- **Commentaires**
  - En pratique: cette approche donne de très bons résultats
  - Principales limitations:
    - il faut apparier les descripteurs d'une image avec tous les descripteurs de la base d'image
      → c'est très lent !
    - Il faut stocker tous les descripteurs locaux des images dans la base
    - 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

- **Approche alternative** (qui sera présentée plus tard dans le cours)
  - Utilisation de **descripteurs globaux** qui agrègent les descripteurs locaux individuels en un descripteur global par image
    - Les descripteurs locaux ne sont plus stockés, seul le descripteur agrégé l'est
  - Le calcul de la similarité entre images se fait par comparaison directe des descripteurs globaux
    - Possibilité d'utiliser des similarités très efficaces