Quelques idées sur le bas niveau en vision par ordinateur: des indices « faits main » aux réseaux convolutionnels

Marie-Odile Berger

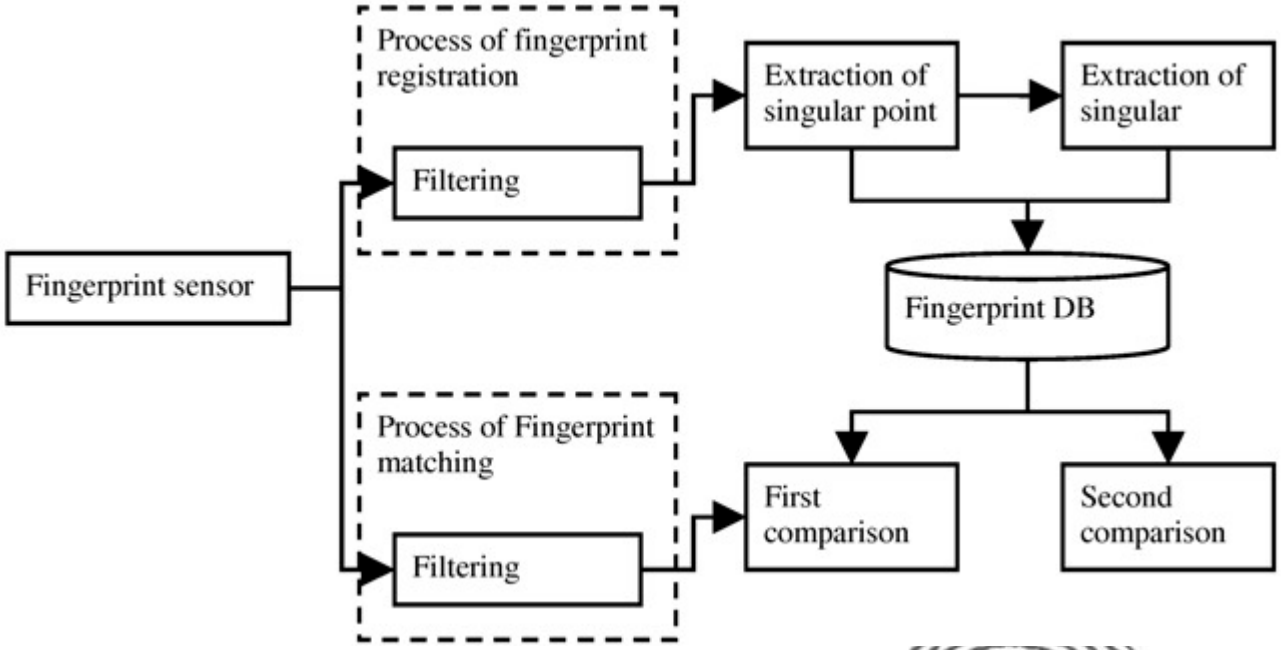
INRIA Nancy Grand Est

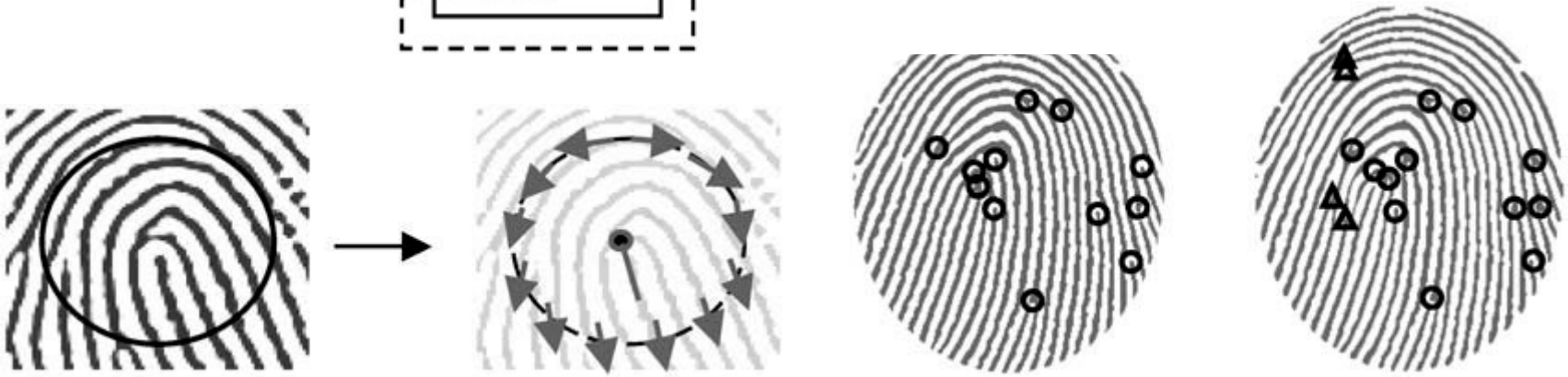
Equipe Tangram

Contact: marie-odile.berger@inria.fr

Page web:http://members.loria.fr/moberger

Un système de vision traditionnel: la reconnaissance d'empreintes





Approches conventionnelles et CNN

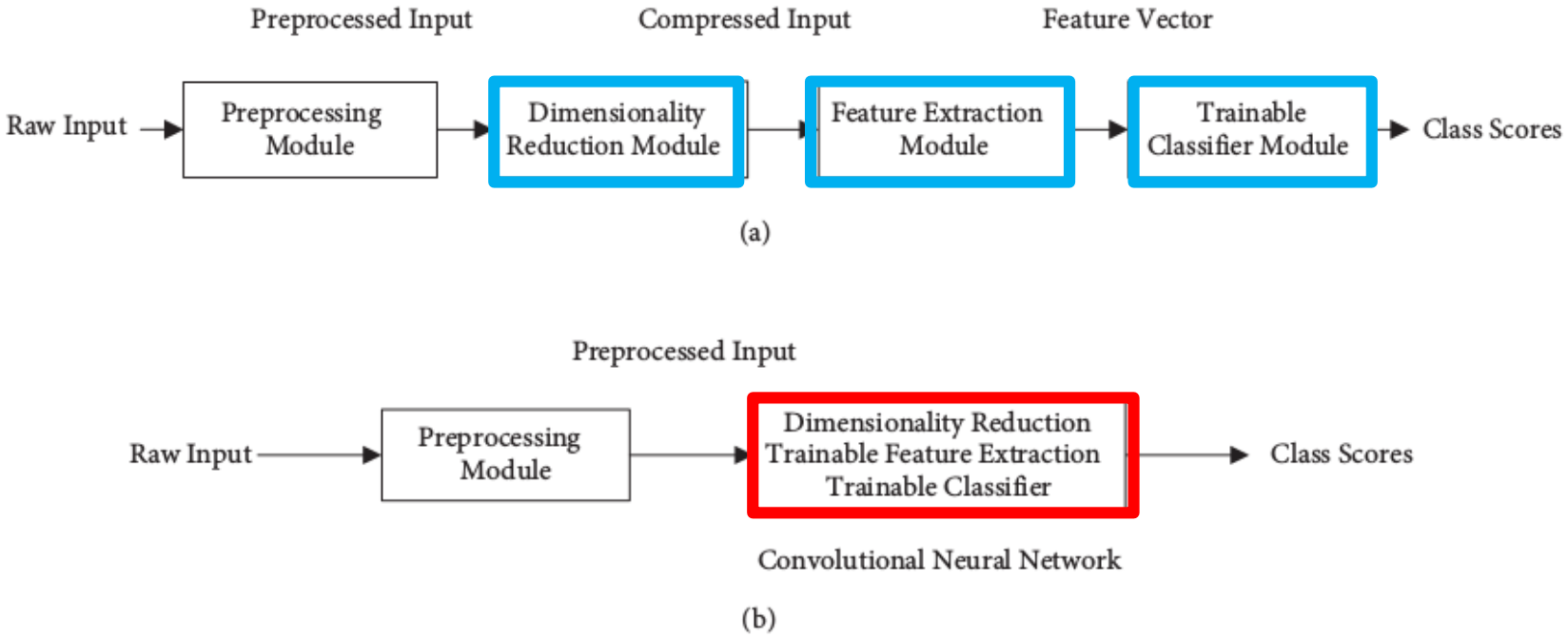


Figure 1. Pattern recognition approaches: (a) conventional, (b) CNN-based.

Approches conventionnelles et CNN

- Dans un **système conventionnel**, le bas niveau est défini **a priori** (« handmade »), indépendamment de la tâche visée
  - et cela peut marcher très bien… (cas du détecteur SIFT, des critères pour les empreintes…)
  - Et cela ne nécessite aucune base de données d'apprentissage

- Dans **un réseau convolutionnel**, les étapes de réduction de la dimensionnalité et de d'extraction des indices de bas niveau sont déterminées lors de l'apprentissage.
  - Le bas niveau est ainsi adapté à un problème donné
  - Mais il faut des données d'apprentissage

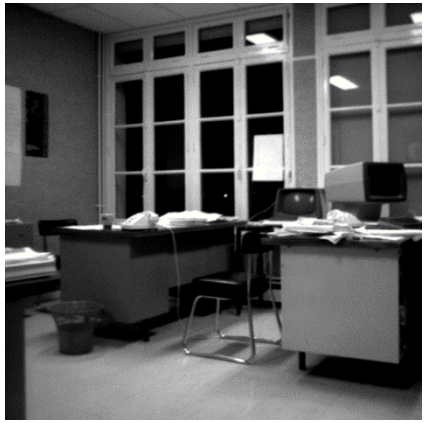
Les indices de bas niveau

Def: Indices extraits dans l'image sans connaissance a priori sur la scène, les objets la composant ou la position de prise de vue

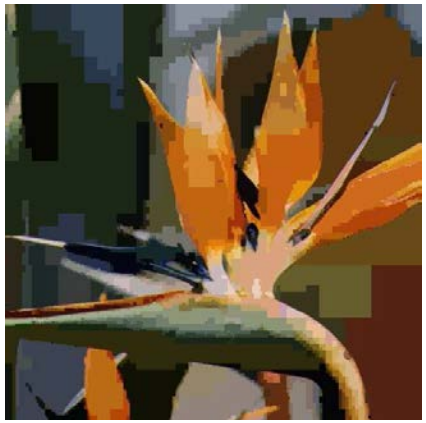
- Contours
- Régions
- Points d'intérêt
- Flot optique (mouvement apparent)

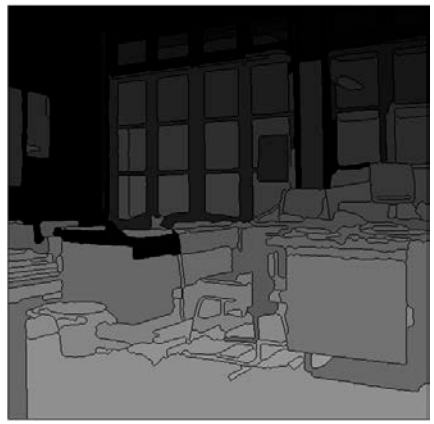
Avec l'arrivée des CNNs, les indices deviennent **de plus haut niveau**. Ce sont aussi des objets identifiables dans les images (piétons, voitures,…)

Exemples d'indices de bas niveau


coins





contours

régions

Qualités souhaitables des indices

Ils ont une sémantique (ex: un contour délimite un objet): en pratique, de nombreux indices sont très efficaces sans avoir de la sémantique (ex points SIFT)

Ils sont largement présents dans les objets qu'on souhaite manipuler

Ils sont (le plus possible) invariants aux conditions d'observation (changements d'illumination, changement de point de vue)

Invariances souhaitables

Point de vue

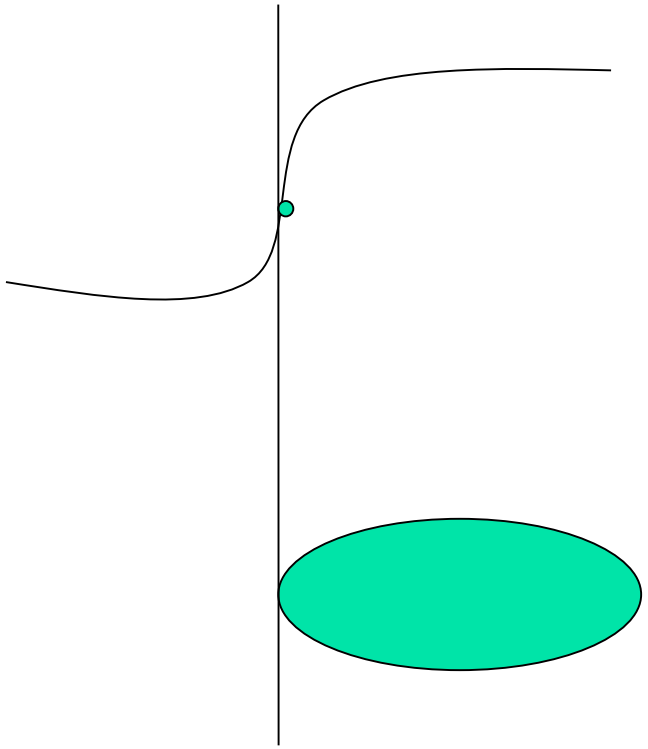
Illumination

Historique du bas niveau

Contour

- Idée contour=**limite** d'un objet
- contour= variation brutale de l'intensité
- Présence d'un **gradient fort** **(Canny 88)**

Région: **zone** d'intensité homogène



Quelques profils d'intensité

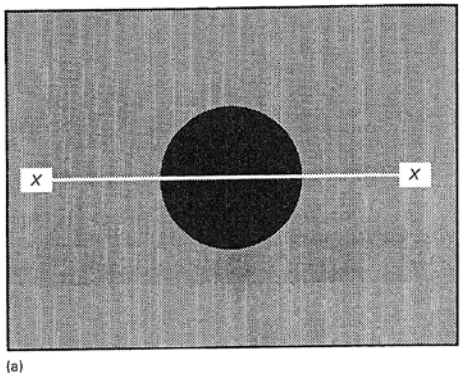
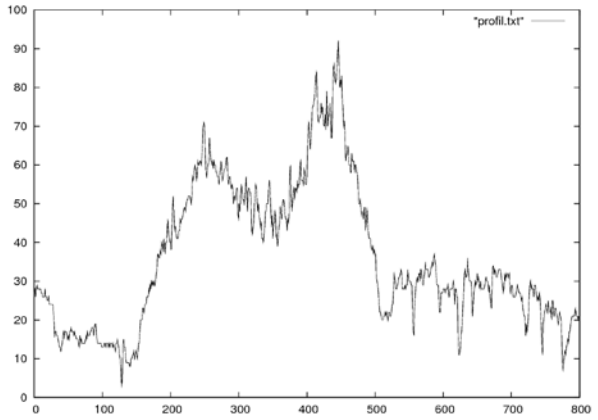


Figure 5.4 Using edge pixels to select a threshold: (a) image of dark, round object on a light background with section $X-X$ shown; (b) profile of image intensity along section $X-X$.



Conclusion:

- la notion de régions avec intensité constante est utopique

Dérivation et bruit ne font pas bon ménage

La dérivation est un problème mal conditionné:

- Une petite perturbation de la fonction initiale cause des perturbations très importantes sur la dérivée.

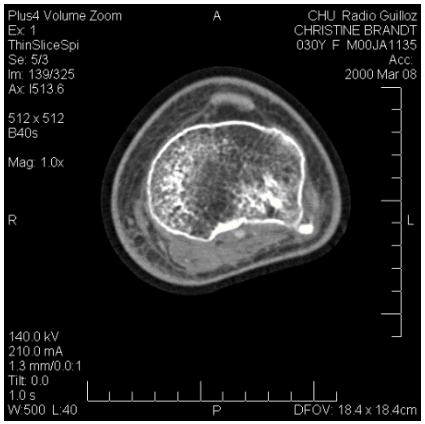
- $f2(t)=f1(t)+ \varepsilon \cos(\omega t))$
- $f'2(t)=f'1(t)- \varepsilon\omega \sin(\omega t))$
- Les fonction sont aussi proches que l'on veut. Pour ω grand, les dérivées peuvent être très éloignées!

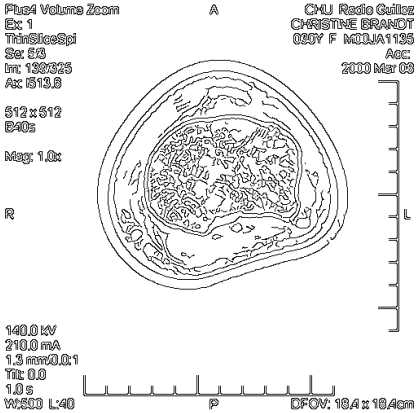
Détection de contour = lisser pour enlever le bruit puis dériver

Dans le cas d'un filtre de convolution

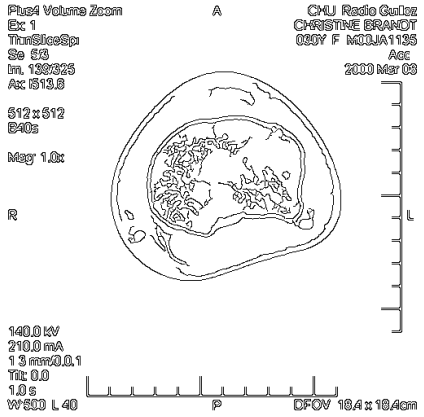
- $(I*G(\sigma))'=I*G'(\sigma))$ convolution avec le filtre dérivé

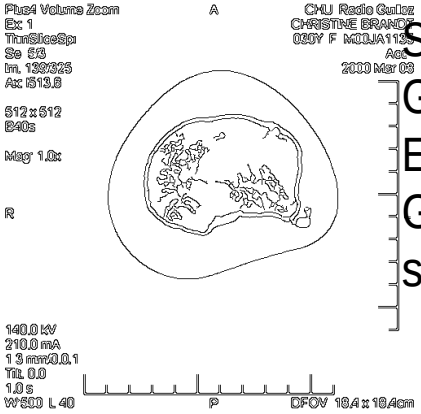
Choisir le seuil sur le module du gradient





Seuil = 10





Seuillage par hysteresis:
Garder tous les points> seuil_haut
Eliminer tous les points < seuil_bas
Garder les points intermédiaires
si la connexité l'impose

Seuil = 20

Seuil = 30

Les points d'intérêt

Extraire des points **sans sémantique** mais très **reproductibles** (i.e. faciles à mettre en correspondance entre deux images)

Utilité:

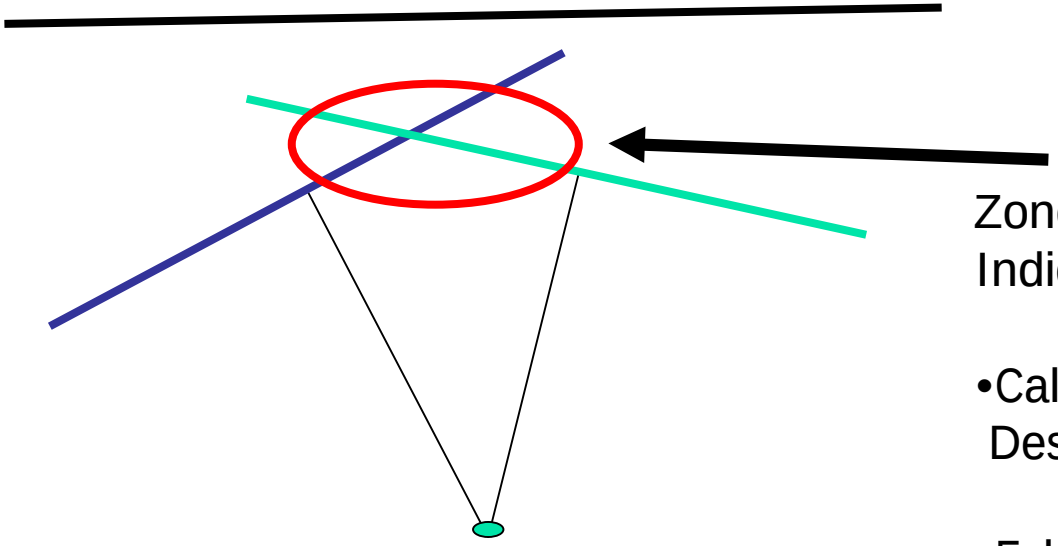
- Information non dense mais sûre
- Permet de calculer des transformations inter-images
- Permet de calculer le point de vue de la caméra à partir de correspondances entre points d'intérêt et points du modèle

Il faut pouvoir **extraire** des points et les **mettre en correspondance**

Exemples d’utilisation des points d’intérêt

Fabriquer des panoramiques

Fabriquer un panoramique

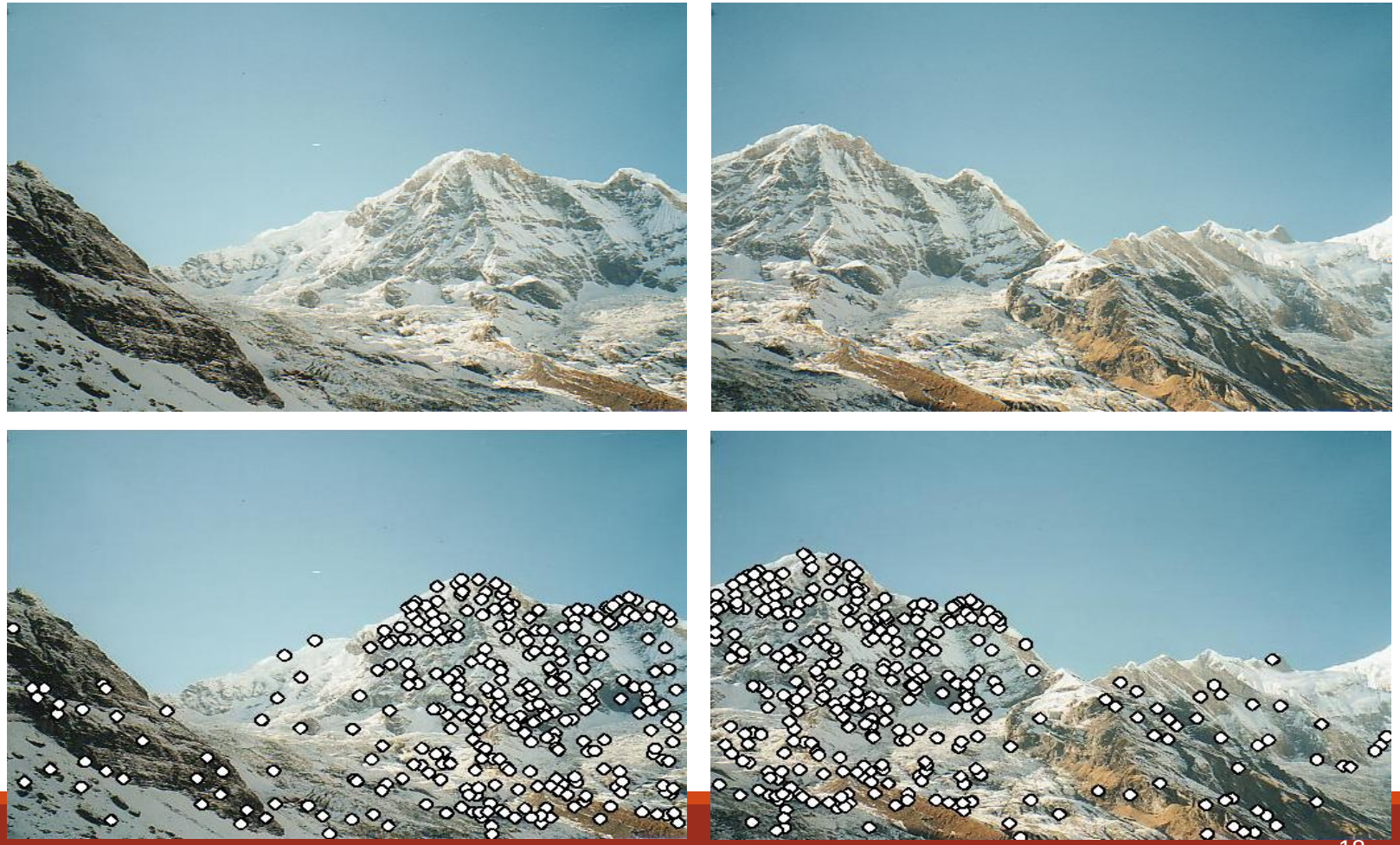


Zone présentant des
Indices communs

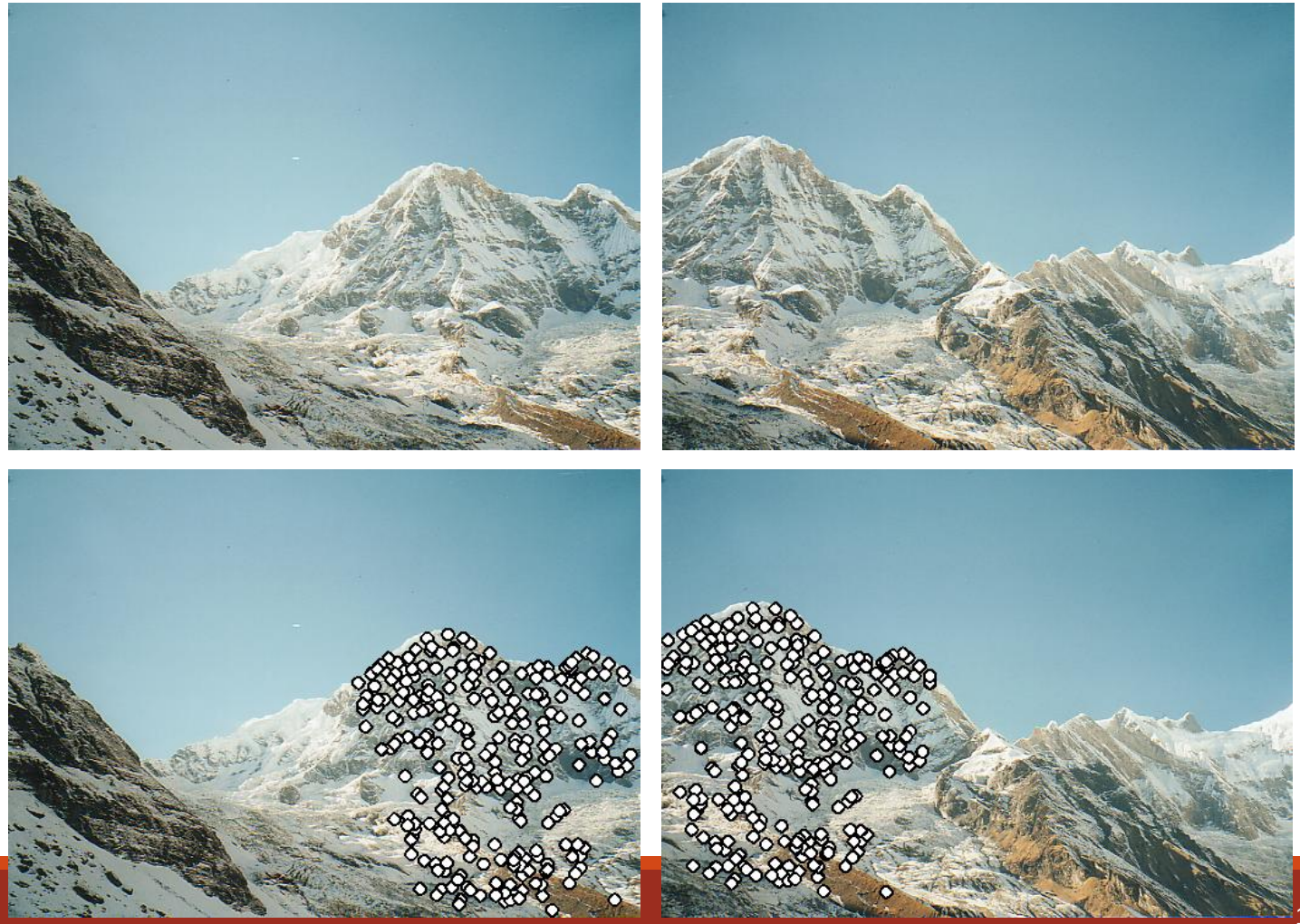
- Calcul des points de vue
  Des caméras

- Fabrication de la nouvelle
  image

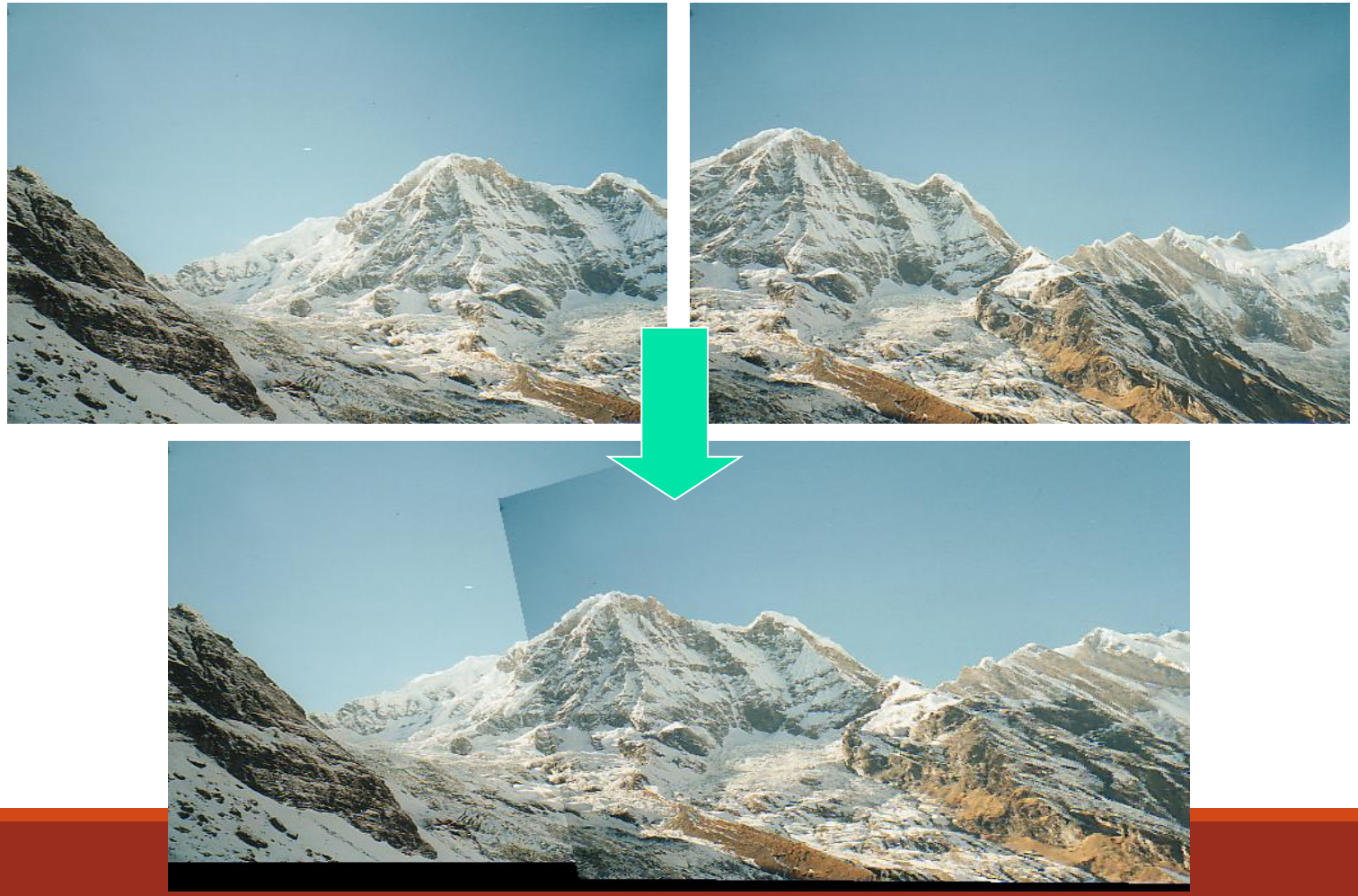
Extraction des points d’intérêt



Détection des points en correspondance (avec l'hypothèse d'homographie)



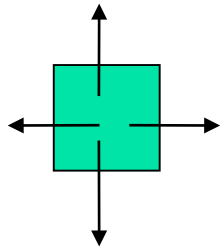
Recollement des images

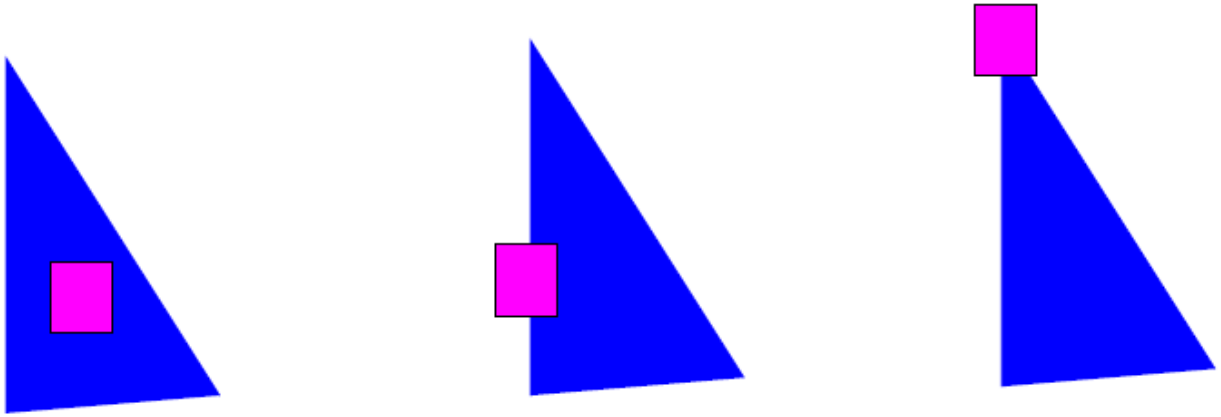


Extraire les points d'intérêt: Les points de Moravec (CMU, 1970…)

Idée de base: observer les changements d'intensité en déplaçant localement une fenêtre:

Point caractéristique = faible auto-similarité





aucun changement

Aucun changement Dans la direction du contour

Coin: changement dans toutes les directions

Points de Moravec

Calculer le changement induit par une translation (u,v) sur une fenêtre décrite par le domaine x,y

$$\sum_{x,y} \left(I(x+u, y+v) - I(x,y) \right)^2$$

Un point intéressant induit **un fort changement**

Directions de changement (u,v) limitées aux axes horizontaux et verticaux

En 1988, points de **Harris**: extension à des directions quelconques u,v

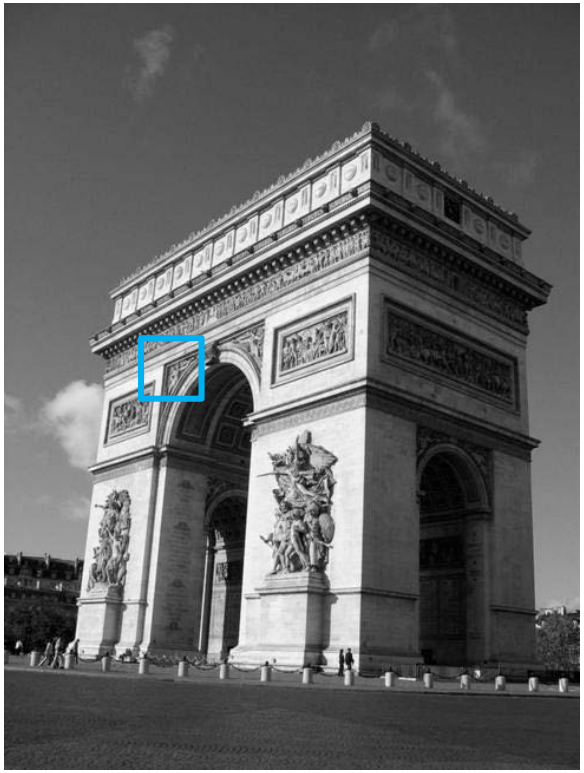
Mise en correspondance

Il ne suffit pas de détecter des points…

Il faut aussi les **mettre en correspondence**

- En se basant sur la ressemblance photométrique au voisinage du point -> construire un **descripteur** associé au point
- Définir **une mesure de similarité** entre descripteurs
- En prenant en compte **l'invariance souhaitée** aux changements d'illumination, d'echelle et d'orientation

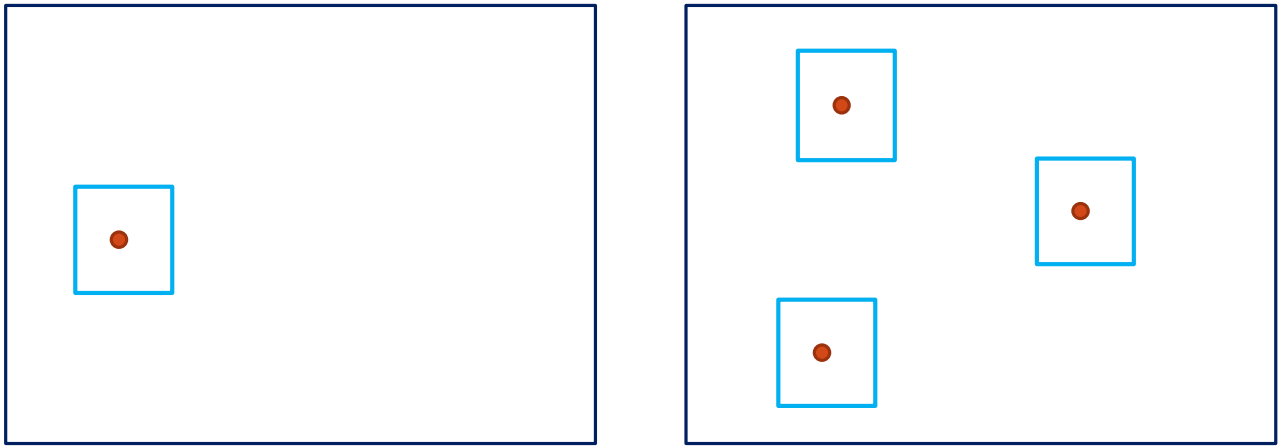
Complexité dans la mise en correspondance



Avec une fenêtre de comparaison de taille constante, forte variation entre fenetres homologues.

Mise en correspondance par corrélation

- Basée sur la ressemblance du voisinage carré des deux points
- Pour un point (x,y) de l'image 1, recherche du point de l'image 2 lui ressemblant le plus



Mise en correspondance par corrélation: critères de similarité

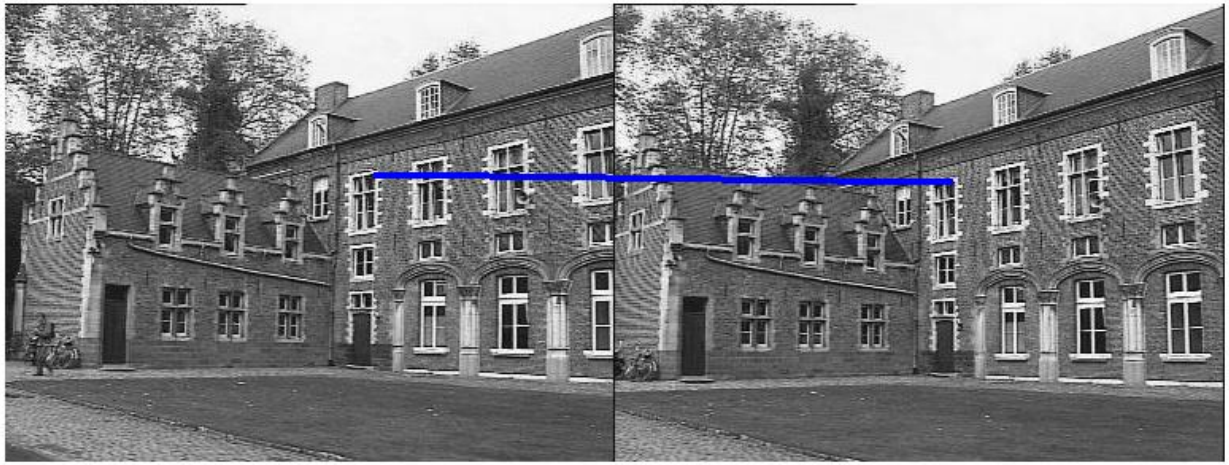
Produit scalaire maximal:

$$cor\big((x,y),(x',y')\big)=\frac{1}{(2w+1)^2}\sum_{i,j=-w}^{i,j=w} I_1(x+i,y+j)\times I_2(x'+i,y'+j)$$

Somme des différences minimale:

$$cor\big((x,y),(x',y')\big)\approx \frac{1}{(2w+1)^2}\sum_{i,j=-w}^{i,j=w}\Big(I_1(x+i,y+j)-I_2(x'+i,y'+j)^2\Big)$$

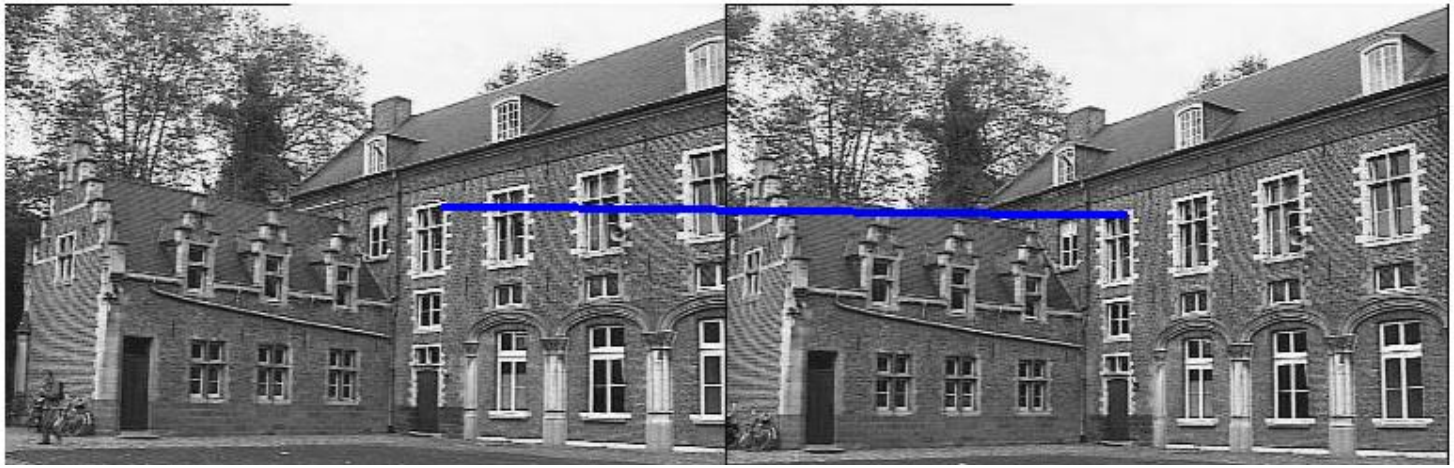
Exemple: point avec la meilleure corrélation avec des points de vues de plus en plus différents



correct

incorrect

Avec une corrélation centrée normalisée



correct

correct

Résoud certains problèmes de variation d'intensité, pas les variations géométriques

SIFT (D. Lowe 2004)

Un détecteur similitude invariant à :

- Scaling
- rotation
- translation

Capable de mettre en correspondance des points distants avec des variations de caméra importantes

Extraire des points d'intérêt

- Considérer $L(x, y, \sigma) = G(\sigma) * I(x, y)$

  Considérer comme points intéressants les extrema de

  $L(x, y, k\sigma) - L(x, y, \sigma)$

- Élimination des points à faible contraste ou localisés sur des contours

- Affecter une orientation aux points d'intérêt: en utilisant le gradient de l'image

  ⇒ un descripteur similitude invariant

Exemple de descripteurs SIFT

Définir un descripteur

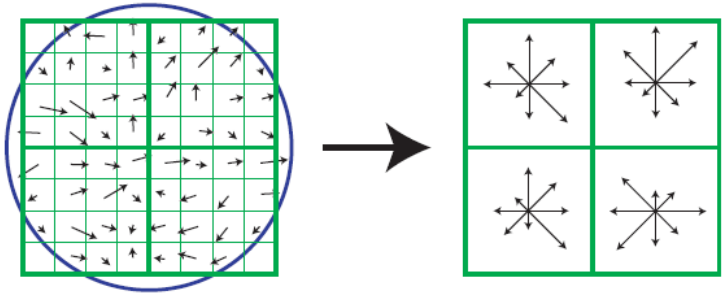
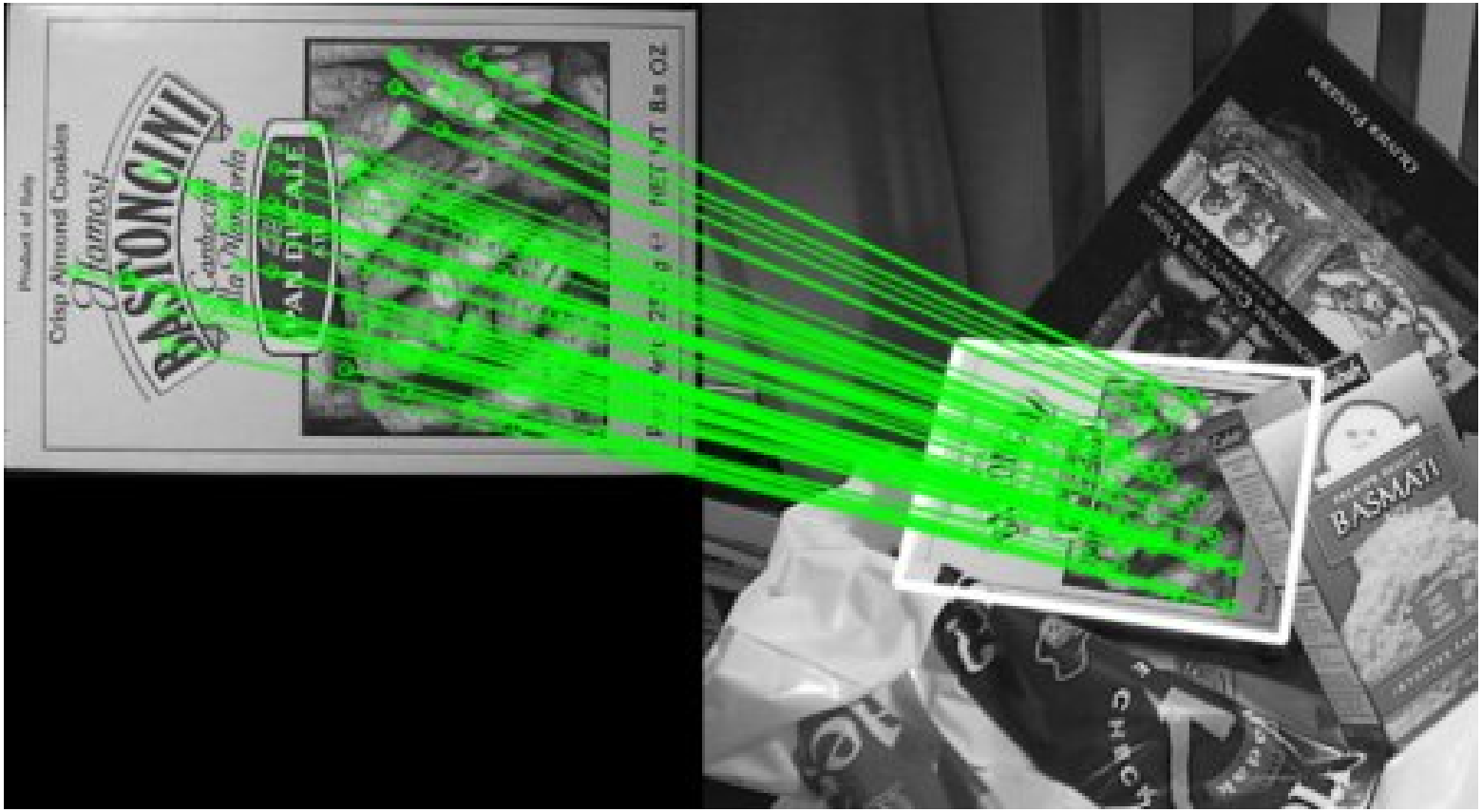


Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

Descripteur= histogramme des gradients calculés **par rapport à l'orientation** du point d'intérêt -> Invariance rotationnelle

En pratique on utilise 16 histogrammes, soit un descripteur à 128 composantes

Mise en correspondance possible malgré les changements de forme et d'orientation

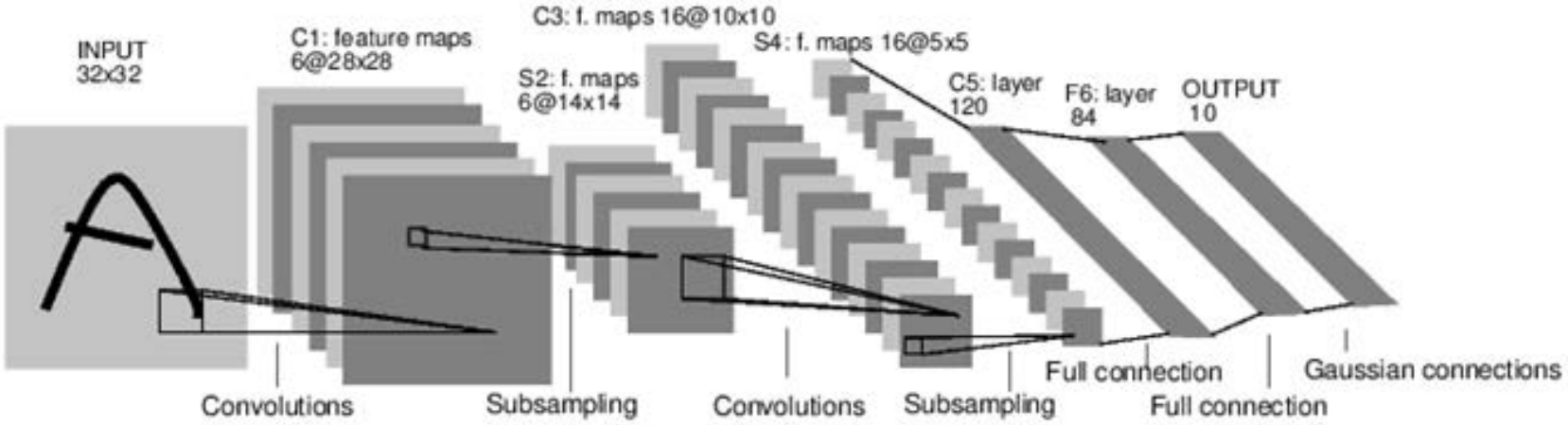


Nouvelles méthodes de détection:

Utiliser les techniques d’apprentissage

Apprentissage et descripteur

Le reconnaissance d'objets à l'aide réseaux convolutionnels a énormément progressé (ex Chopra 2005)



A Full Convolutional Neural Network (LeNet)

Détermination des filtres de façon à minimiser l'écart entre vérité terrain et Prédiction sur l'ensemble des données d'apprentissage.

Les descripteurs CNN

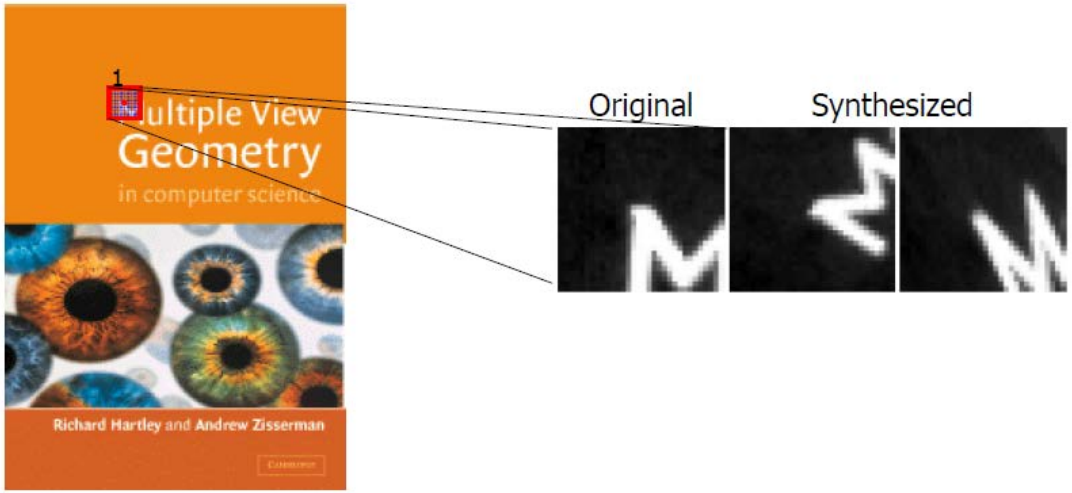
- La dernière couche (avant la classification) a permis de bien classifier les données:
  - C'est donc a priori un bon candidat pour décrire une forme : c'est **le descripteur CNN** (convnet landmarks)
- des descripteurs (assez) *génériques* proviennent de réseaux comme Alexnet ou googleLeNet, utilisés pour la classification et entrainés sur de **très grandes bases de données** (ImageNet)
  - Souvent utilisés pour trouver l'image la plus proche d'une image requête(voir le TD)
  - Bonne invariance aux changements de condition (lumière, saison…)

Mise en correspondance de points et apprentissage

Principe général: fournir au **réseau des exemples d'un même point physique** vu sous des angles différents et dans des conditions différentes (illumination notamment)

Une classe= les différentes apparences d'un même point physique

Mettre en correspondance des points= identifier leur classe (Lepetit 2004)



Correspondances et réseaux siamois

Les réseaux siamois sont à la base des méthodes de mise en correspondances de patchs. L'objectif est d'apprendre **une métrique de similarité**

1. Principe: fournir en entrée des paires de patchs en correspondance (stereo, structure from motion) et des paires négatives
2. Soumettre les deux patchs au même réseau
3. Le réseau est optimisé pour fournir une valeur petite pour les données ressemblantes et élevée pour les paires négatives

Architecture des réseaux siamois

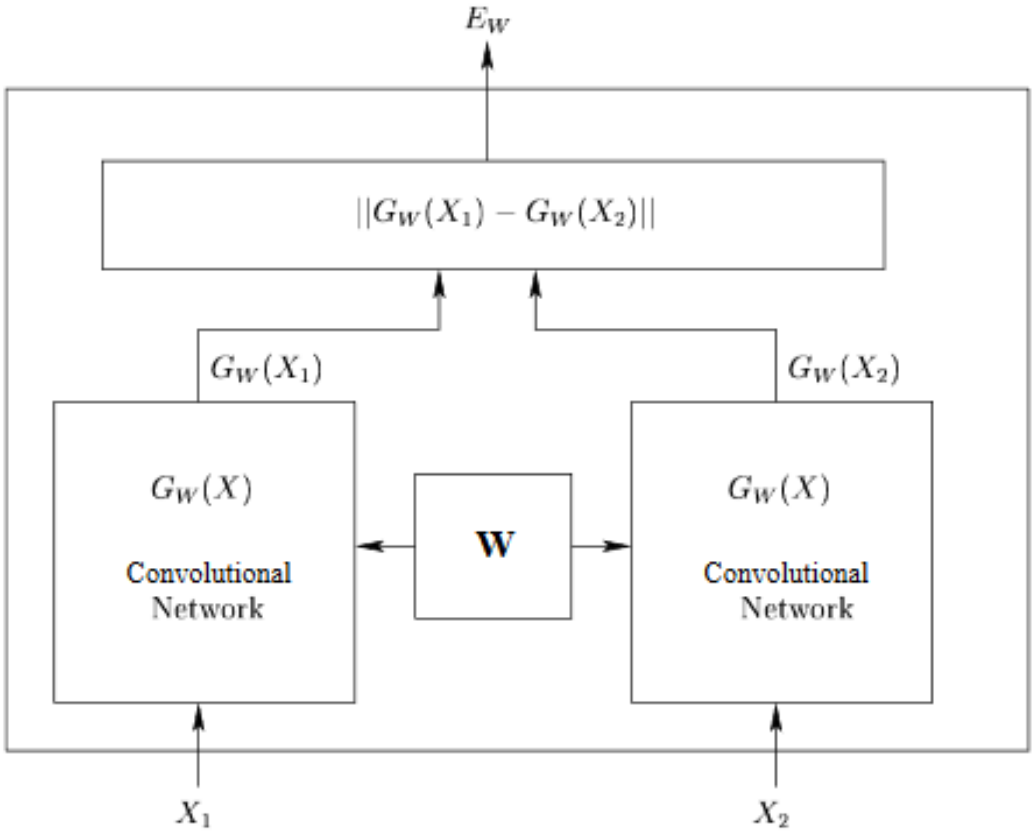


Figure 1. Siamese Architecture.

X1 et X2 semblable: Y=0
Sinon Y=1 (imposteur)

Fonction de perte:
$(1-Y)E\_w^2 + Y\exp(-Ew)$

Chopra, Lecun 2005

Exemple: LIFT [Moo2016]

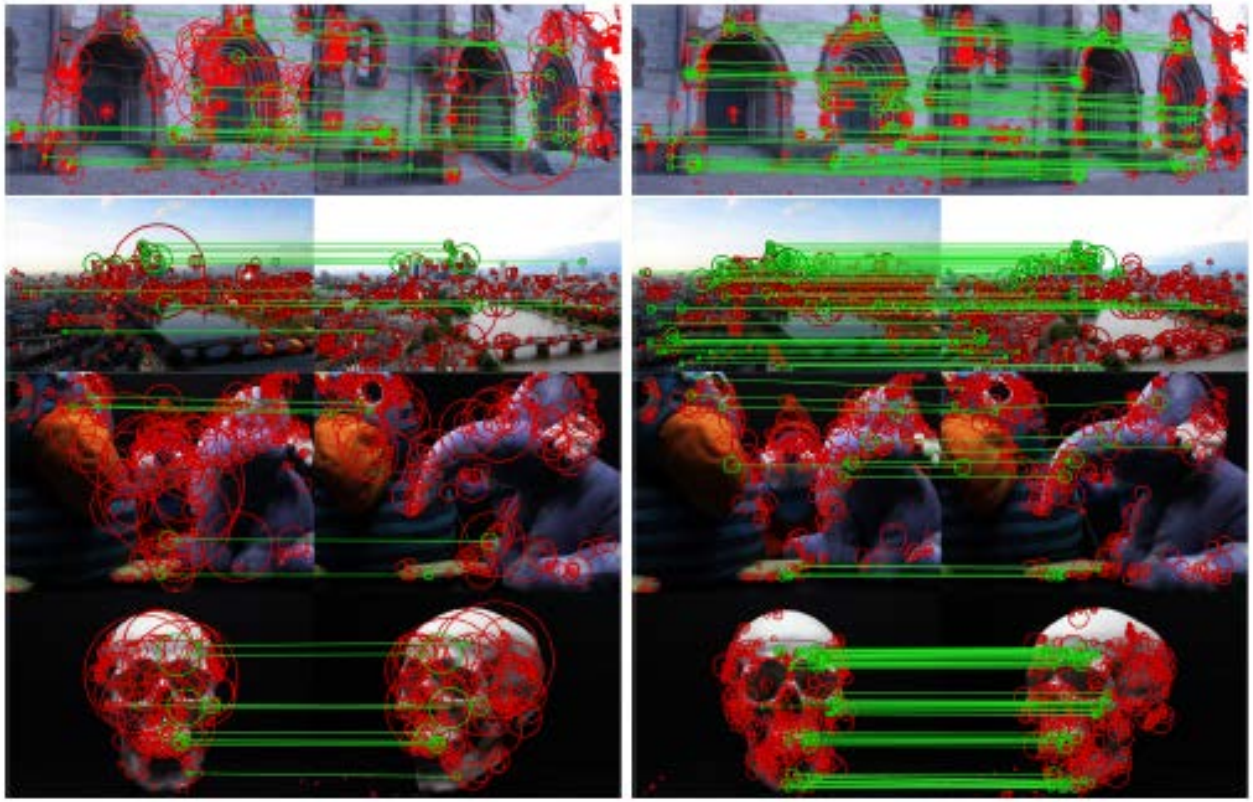


Fig. 5. Qualitative local feature matching examples of **left:** SIFT and **right:** our method LIFT. Correct matches recovered by each method are shown in green lines and the descriptor support regions with red circles. **Top row:** *Herz-Jesu-P8* of *Strecha*, **second row:** *Frankfurt* of *Webcam*, **third row:** *Scene 7* of *DTU* and **bottom row:** *Scene 19* of *DTU*. Note that the images are very different from one another.

Bibliographie

Canny J. : A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(8), 1986.

Lindeberg T.: Scale-space for Discrete Signals. IEEE Transactions on Pattern Analysis and Machine Intelligence, 3(12), 1990

Kass M., Witkin A., Terzopolos D.: Snakes: Active contour models. International Journal on Computer Vision, 1988.

Harris C., Stephens M.: A Combined Corner and Edge Detector. 1988

Lowe D.: Distinctive image features from scale-invariant Keypoints. International Journal on Computer Vision. 60 (2), 2004.

Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(2004) 1615–1630

Bibliographie

Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Surf: Speeded Up Robust Features. Computer Vision and Image Understanding 10(2008) 346–359

Calonder, M., Lepetit, V., Konolige, K., Bowman, J., Mihelich, P., Fua, P.: Compact Signatures for High-Speed Interest Point Description and Matching. In: International Conference on Computer Vision. (2009)

Heinly J., Dunn E., Frahm J.M. Comparative Evaluation of Binary features. In ECCV 2012

Lepetit, Pilet, Fua: PointMatching as a Classification Problem for Fast and Robust Object Pose Estimation, CVPR 2004

Rublee E., Rabaud V., Konolige K., Bradski G: ORB: An efficient alternative to SIFT or SURF. ICCV 2011: 2564-2571

Correspondance et apprentissage

Chopra, Hadsell, Le Cun, Learning a Similarity Metric Discriminatively, with Application to Face Verification, CVPR 2005

X. Han, T. Leung, Y. Jia, R. Sukthankar, A. Berg: MatchNet: Unifying feature and metric learning for patch-based matching, *CVPR* (*2015*)

LIFT: Learned Invariant Feature Transform. Moo Yi, Eduard Trulls, Vincent *Lepetit*, Pascal Fua, ECCV 2016

A. Zamir, T. Wekel, P. Agrawal, C. Weil, J. Malik, S. Savarese. Generic 3D Representation via Pose Estimation and Matching. ECCV 2016

Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, Pascal Fua. LIFT: Learned Invariant Feature Transform. CVPR 2016