

Traitement Avancé des Images

Primitives

T. Dietenbeck

thomas.dietenbeck@sorbonne-universite.fr



- 1 Introduction
- 2 Détection de points d'intérêt
- 3 Description des points d'intérêt
- 4 Matching de points d'intérêt

1 Introduction

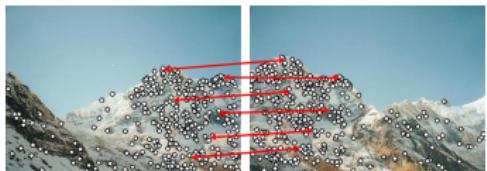
- 2 Détection de points d'intérêt
- 3 Description des points d'intérêt
- 4 Matching de points d'intérêt

Objectifs

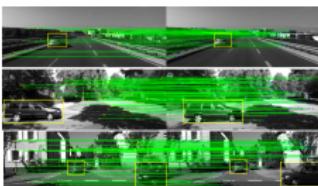
- Résumer une image / un objet par quelques points caractéristiques

Applications

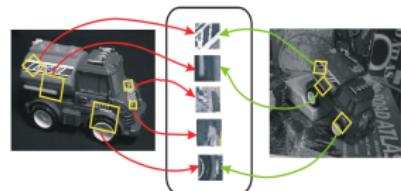
- Panorama, stéréo, reconstruction 3D
- Suivi de mouvement
- Reconnaissance d'objets
- Robotique (navigation)
- ...



Panorama



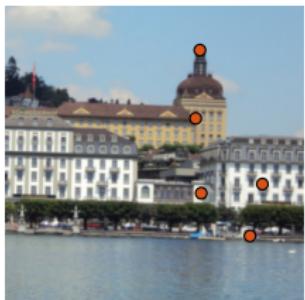
Suivi de mouvement



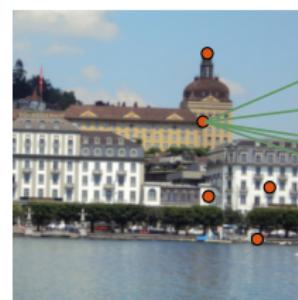
Reconnaissance d'objets

Caractéristiques

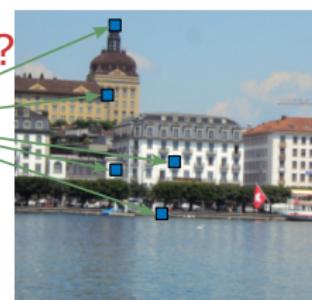
- **Répétable** : un point doit pouvoir être trouvé dans une image malgré des transformations géométriques ou photométriques
- **Local** : un point doit occuper / correspondre à une petite portion de l'image et être robuste à l'occlusion ou à l'encombrement
- **Saillant** : chaque point doit être distinct des autres
- **Compact et efficace** : beaucoup moins de points d'intérêt que de pixels dans l'image (pour permettre des calculs rapides)



Répétabilité

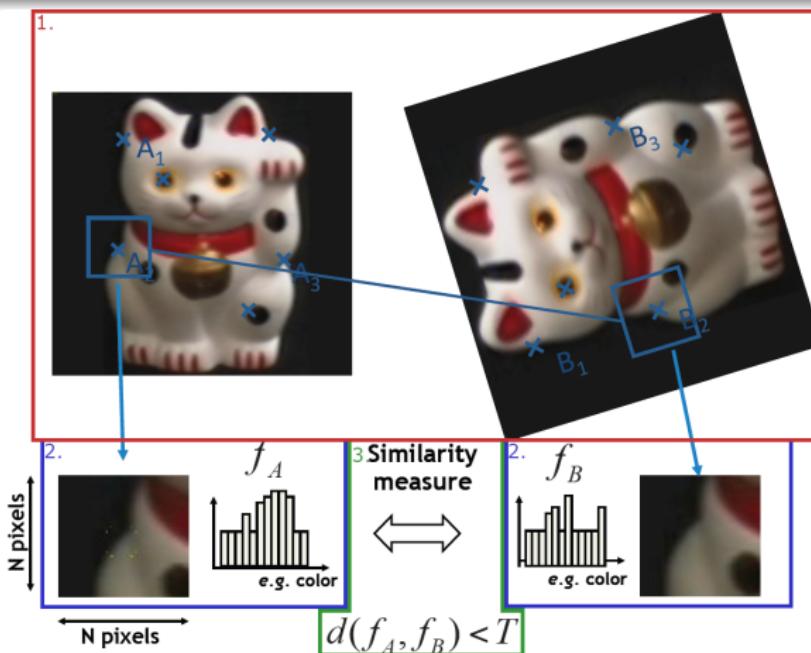


Saillance



Principes / Étapes

- ① Déetecter un ensemble de points d'intérêts (key-points)
- ② Décrire la région les entourant
- ③ Faire correspondre les descripteurs de 2 images



Quelques méthodes existantes

• Détection

- Détecteur de Harris [Harris, Alvey Vision Conf., 1988]
- Matrice hessienne, blobs, ...
- Smallest Univalue Segment Assimilating Nucleus (SUSAN) [Smith, Intl. J. Comp. Vis., 1997]
- Features from Accelerated Segment Test (FAST) [Rosten, Pattern Anal. and Machine Intel., 2008]

• Description

- Bloc d'intensité
- Histogram of Gradient (HoG) [Lowe, Intl. J. Comp. Vis., 2004]
- Local Binary Pattern (LBP) [Ojala, Pattern Recog., 1996]

• Détection et description

- Scale Invariant Feature Tracking (SIFT) [Lowe, Intl. J. Comp. Vis., 2004]
- Speeded-Up Robust Features (SURF) [Bay, ECCV, 2006]
- Binary Robust Independent Elementary Features (BRIEF) [Calonder, Pattern Anal. and Machine Intel., 2012]
- Oriented Fast and Rotated BRIEF (ORB) [Rublee, ICCV, 2011]
- Binary Robust Invariant Scalable Keypoints (BRISK) [Leutenegger, ICCV, 2011]
- (Accelerated) KAZE [Alcantilla, Pattern Anal. and Machine Intel., 2011]

1 Introduction

2 Détection de points d'intérêt

- Détection d'un coin [Harris, 1988]
- Intégration de l'échelle
- Détection de blobs

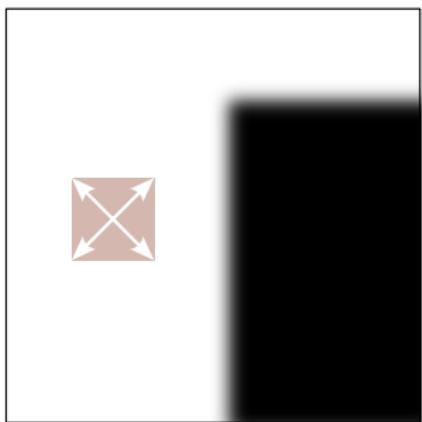
3 Description des points d'intérêt

4 Matching de points d'intérêt

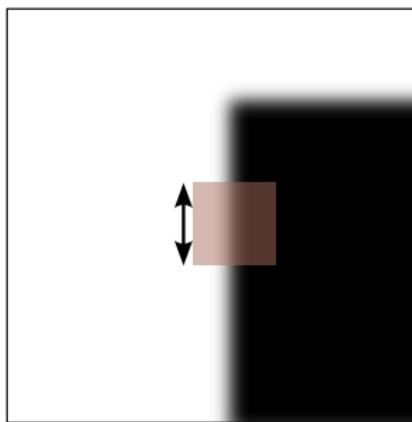
Détection d'un coin [Harris, 1988]

Idée de départ

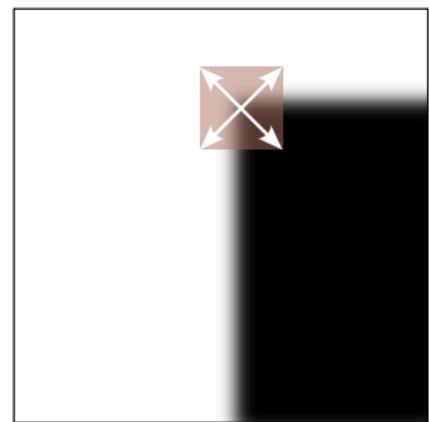
- La localisation d'un coin doit pouvoir se faire à l'aide d'une (petite) fenêtre
- Déplacer la fenêtre dans n'importe quelle direction doit entraîner de larges changements d'intensité.



Région homogène : pas de changement d'intensité



Contour : pas de changement d'intensité suivant le contour



Coin : changement d'intensité dans toutes les directions

Détection d'un coin [Harris, 1988]

Idée de départ

- La localisation d'un coin doit pouvoir se faire à l'aide d'une (petite) fenêtre
- Déplacer la fenêtre dans n'importe quelle direction doit entraîner de larges changements d'intensité.

Les maths

- Décalage $\vec{V} = (u, v)$ de la fenêtre $W \Rightarrow$ Changement d'intensité

$$E(u, v) = \sum_{(x,y) \in W} (I(x+u, y+v) - I(x, y))^2$$

- Petit déplacement \Rightarrow Développement de Taylor (*cf. flux optique et estimation de mouvement*) : $I(x+u, y+v) = I(x, y) + I_x u + I_y v$

$$E(u, v) = \sum_{(x,y) \in W} (I_x u + I_y v)^2 = \vec{V} \mathbf{M} \vec{V}^T$$

$$\boxed{\mathbf{M} = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix} = \mathbf{R}^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \mathbf{R}}$$

Détection d'un coin [Harris, 1988]

Critère de Harris

$$\mathbf{M} = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix} = \mathbf{R}^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \mathbf{R}$$

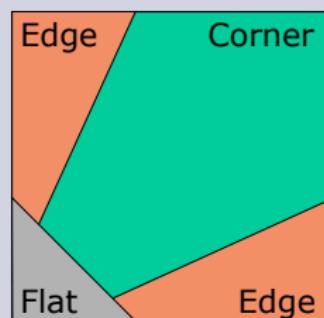
$$C = \det(\mathbf{M}) - k \text{trace}(\mathbf{M})^2$$

- $k \in [0.04; 0.06]$ généralement
- Permet de comparer les valeurs propres de la matrice sans les calculer !

Interprétation

- **Coin** : $C > 0 \Rightarrow \lambda_1 \simeq \lambda_2$ et λ_i grand
- **Contour** : $C < 0 \Rightarrow \lambda_1 \gg \lambda_2$ ou $\lambda_1 \ll \lambda_2$
- **Région homogène** : $|C| \simeq 0 \Rightarrow \lambda_1 \simeq \lambda_2 \simeq 0$

Même principe / idée que l'algorithme de Lucas & Kanade !



Détection d'un coin [Harris, 1988]

Critère de Harris

$$\mathbf{M} = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix} = \mathbf{R}^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \mathbf{R}$$

$$C = \det(\mathbf{M}) - k \text{trace}(\mathbf{M})^2$$

- $k \in [0.04; 0.06]$ généralement
- Permet de comparer les valeurs propres de la matrice sans les calculer !

Remarques

Tout comme pour l'algorithme de Lucas & Kanade, on lisse / pondère généralement les dérivées par une fenêtre gaussienne (plutôt que rectangle)

$$\mathbf{M} = g(\sigma) * \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

- $g(\sigma)$ est un filtre gaussien d'écart-type σ
- La convolution fait déjà une somme pondérée (donc pas besoin de la faire apparaître dans la matrice)
- La pondération gaussienne rend le détecteur invariant en rotation

Matrice hessienne

- Même principe que le détecteur de coin de Harris mais on utilise la matrice hessienne $\mathbf{H} = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}$ au lieu de la matrice \mathbf{M} .
- Critère : $C = \det(\mathbf{H})$
- Les coins correspondent aux maxima de C

Remarque

- Le détecteur est invariant en rotation

Suppression des non-maxima locaux

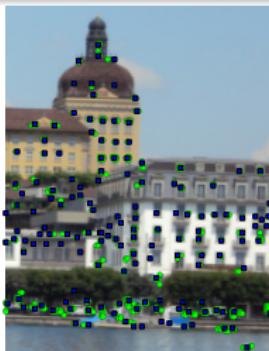
Problème

Les points autour d'un coin vérifient souvent également les critères de sélection

- ⇒ ils sont généralement détectés et provoquent des "amas"
- ⇒ beaucoup de points proches peuvent entraîner de mauvais matchings (et beaucoup plus de temps de calcul)

Suppression des non-maxima locaux

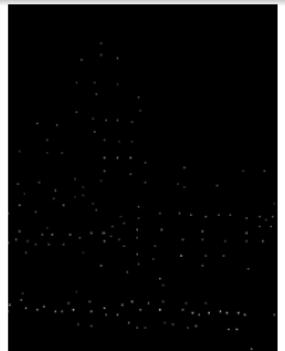
- Avant de seuiller le critère (et d'en déduire les points d'intérêt), on met à 0 tous les points qui n'ont pas la valeur maximale dans un voisinage.



Points détectés avant (vert) et après (bleu) suppression



Scores avant suppression



Scores après suppression

Invariances

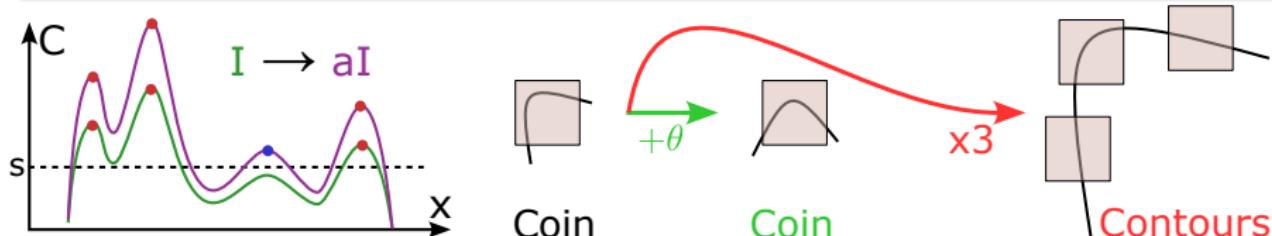
Les détecteurs de coin de Harris et basé sur la matrice hessienne sont invariants

- en translation (évident puisqu'on parcourt toute l'image)
- en rotation (à condition d'utiliser une fenêtre gaussienne pour Harris)
- à un shift d'intensité ($I \mapsto I + b$) : utilisation des gradients et donc de différences d'intensité

Problème

Ces 2 détecteurs ne sont pas invariants

- à un changement de contraste ($I \mapsto aI$)
- à la mise à l'échelle

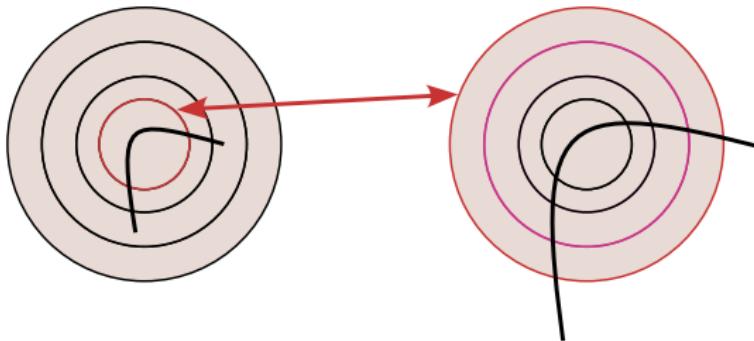


Question

- Comment rendre un détecteur invariant à une mise à l'échelle ?

Idées

- Utiliser plusieurs résolutions d'image
- Considérer des régions de tailles différentes autour d'un point

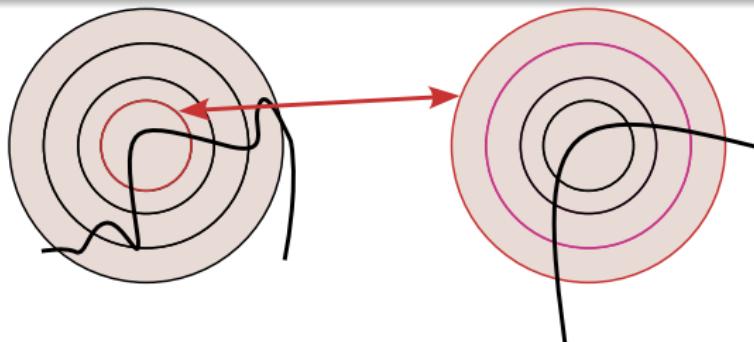


Question

- Comment rendre un détecteur invariant à une mise à l'échelle ?

Idées

- Utiliser plusieurs résolutions d'image
- Considérer des régions de tailles différentes autour d'un point



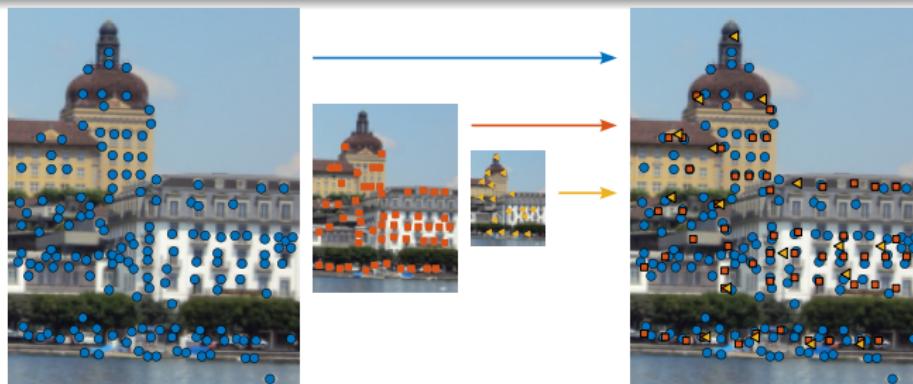
Questions

- Peut-on trouver / identifier aisément la bonne échelle pour un objet ?
- Comment choisir les tailles / échelles indépendamment pour chaque image ?

Intégration de l'échelle et détection de coin

Pyramide de multi-résolution

- ① On sous-échantillonne plusieurs fois l'image pour créer une "pyramide"
- ② On effectue la détection sur chaque image
- ③ On conserve tous les coins détectés et la résolution correspondante (pour la description)



Problème

- Un même point (ou un point proche) peut être détecté à plusieurs échelles
⇒ amas de points (et augmentation des temps de calcul)

Intégration de l'échelle et détection de coin

SUSAN [Smith, Intl. J. Comp. Vis., 1997]

Pour chaque pixel \mathbf{p}_0 de l'image,

- Considérer un cercle \mathcal{C} de rayon r et centré sur \mathbf{p}_0

- Critère : $C(\mathbf{p}_0) = \max(t - R(\mathbf{p}_0), 0)$

$$\text{avec } R = \sum_{\mathbf{p} \in \mathcal{C}} \begin{cases} 1 & \text{si } |I(\mathbf{p}) - I(\mathbf{p}_0)| \leq g \\ 0 & \text{sinon} \end{cases} \text{ et } g, t \text{ des seuils à fixer}$$

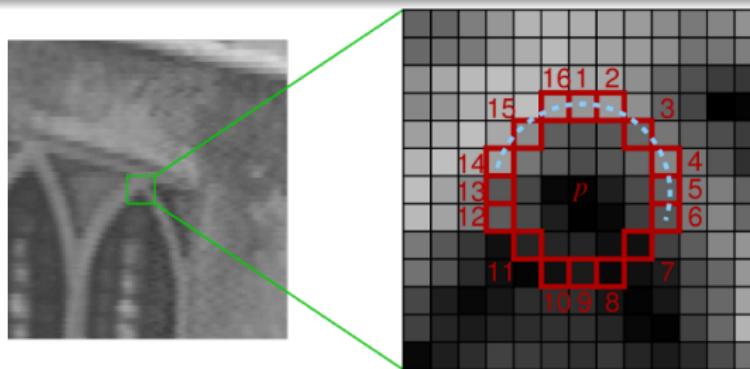
Problèmes

- Beaucoup de calcul pour chaque pixel (et chaque valeur de r)
- Comment choisir les valeurs de seuil ?

Intégration de l'échelle et détection de coin

FAST [Rosten, Pattern Anal. and Machine Intel., 2008]

- Pour chaque pixel p_0 de l'image, considérer les 16 pixels sur un cercle \mathcal{C} de rayon 3 et centré sur p_0
- p_0 est un coin si n pixels consécutifs du cercle ont une intensité supérieure à $I(p_0) + t$ (**OU** une intensité inférieure à $I(p_0) - t$)
- n et t sont des paramètres à fixer ; t est empiriquement fixé à 9 et $n > 8$

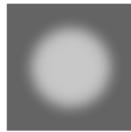


Problèmes

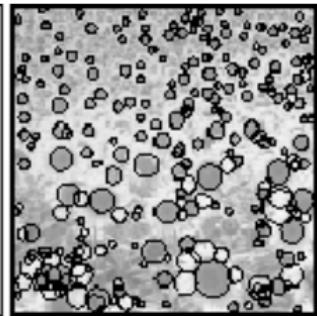
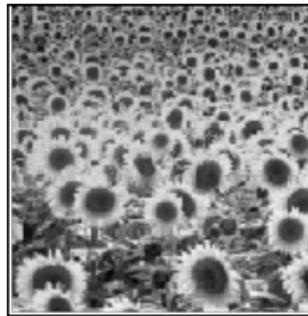
- Pas d'invariance à l'échelle
- Comment choisir les valeurs de seuil ?

Définition

Un blob est une zone homogène de l'image entourée de pixels d'une autre couleur / intensité.



Blob



Intérêts

- Détection du centre de l'objet (et non de ses coins)
- Facile d'estimer la taille / échelle optimale (contrairement à un coin)

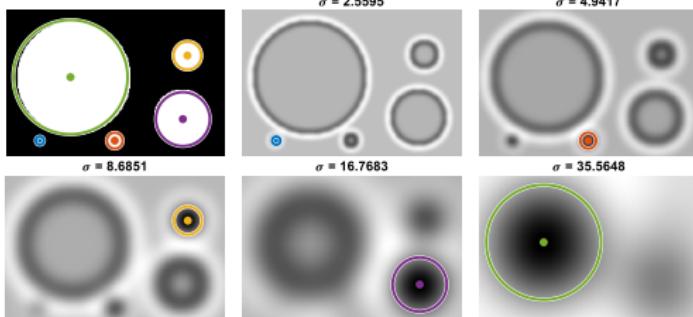
Détection d'un blob et de son échelle

- Un blob peut facilement être détecté par un filtre Laplacien de Gaussienne (LoG)

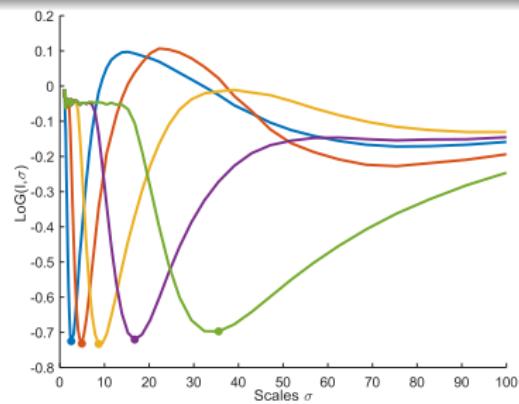
$$LoG_\sigma(x, y) = \frac{-1}{\pi\sigma^2} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) \exp^{-\frac{x^2+y^2}{2\sigma^2}} = -2 \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) G_\sigma(x, y)$$

avec $G_\sigma(x, y)$ la Gaussienne 2D d'écart-type σ

- L'échelle optimale pour un point correspond à la valeur de σ ($= r/\sqrt{2}$) donnant l'amplitude maximale (ou minimale) du LoG.



Détection du centre des cercles



Détection de l'échelle optimale

Détection d'un blob et de son échelle

- Un blob peut facilement être détecté par un filtre Laplacien de Gaussienne (LoG)

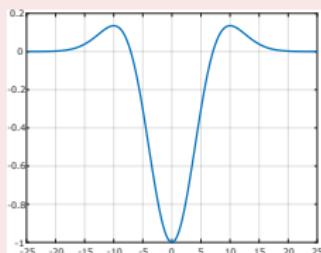
$$LoG_\sigma(x, y) = \frac{-1}{\pi\sigma^2} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) \exp^{-\frac{x^2+y^2}{2\sigma^2}} = -2 \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) G_\sigma(x, y)$$

avec $G_\sigma(x, y)$ la Gaussienne 2D d'écart-type σ

- L'échelle optimale pour un point correspond à la valeur de σ ($= r/\sqrt{2}$) donnant l'amplitude maximale (ou minimale) du LoG.

Remarques

- Un blob clair sur fond sombre correspondra à un minimum du LoG
 - Un blob sombre sur fond clair correspondra à un maximum du LoG
- ⇒ il est préférable d'utiliser $|LoG * I|$



Problèmes

- L'échelle est une variable continue \Rightarrow quelles valeurs de σ choisir pour le filtre ?
- Le calcul pour ces différentes échelles peut être coûteux :
 - en mémoire : il faut stocker toutes les images filtrées (pour la description des points d'intérêt)
 - en temps : il faut calculer toutes les convolutions (filtres de grandes tailles ($6\sigma, 6\sigma$))

Solutions

- log-scale space
- Pyramide de résolution et octaves
- Différences de Gaussiennes (DoG)

log-scale space et octaves

- On choisit n_s valeurs $\sigma_i = 2^{\frac{i}{n_s}} \sigma_0 = k^i \sigma_0 \Rightarrow$ échelle / scale logarithmique
- Les images obtenues par filtrage avec ces valeurs σ_i forment une octave ($\sigma_{n_s} = 2\sigma_0$)
- La dernière image d'une octave est sous-échantillonnée d'un facteur 2 et sert à la création d'une nouvelle octave (avec les mêmes valeurs de σ_i !)

Propriété

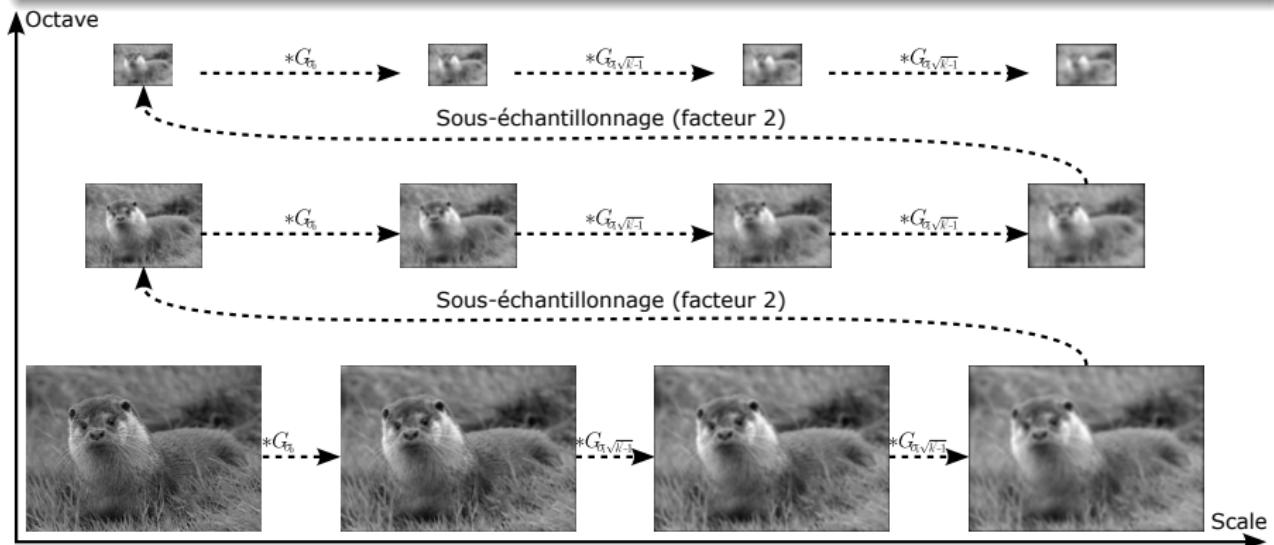
- Les convolutions Gaussiennes forment un semi-groupe : $G_{\sigma_1} * G_{\sigma_2} = G_{\sqrt{\sigma_1^2 + \sigma_2^2}}$
- $$\Rightarrow G_{\sigma_{i+1}} * I = G_{\sigma_i \sqrt{k^2 - 1}} * (G_{\sigma_i} * I) \text{ car } \sigma_{i+1} = k\sigma_i = \sqrt{\sigma_i^2 + \sigma_i^2(k^2 - 1)}$$

Intérêts

- Plus on monte en octave, plus les images sont petites \Rightarrow moins de temps de calcul et de mémoire pour conserver les images
- Les filtres $G_{\sigma_i \sqrt{k^2 - 1}}$ sont plus petits que les filtres $G_{k\sigma_i}$ \Rightarrow moins de temps de calcul pour les convolutions

log-scale space et octaves

- On choisit n_s valeurs $\sigma_i = 2^{\frac{i}{n_s}} \sigma_0 = k^i \sigma_0 \Rightarrow$ échelle / scale logarithmique
- Les images obtenues par filtrage avec ces valeurs σ_i forment une octave ($\sigma_{n_s} = 2\sigma_0$)
- La dernière image d'une octave est sous-échantillonnée d'un facteur 2 et sert à la création d'une nouvelle octave (avec les mêmes valeurs de σ_i !)

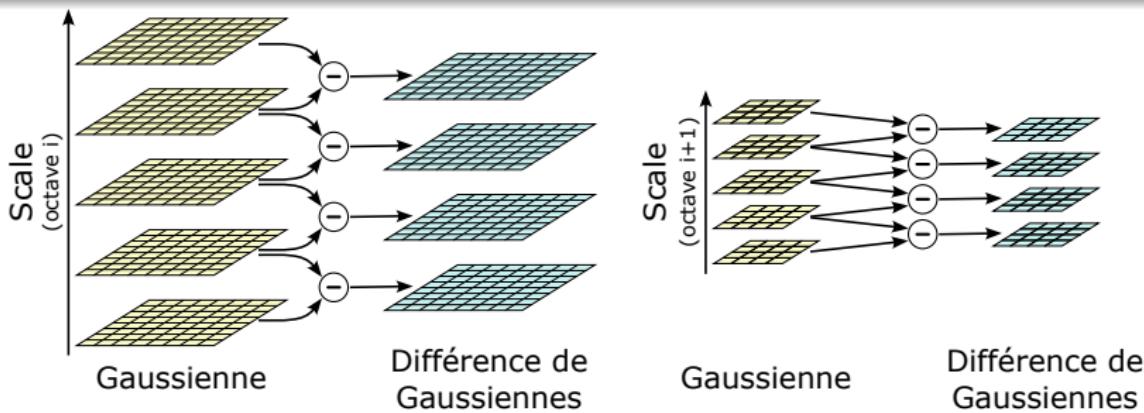


Différences de Gaussiennes (DoG)

- Une bonne approximation du LoG peut être obtenue par la différence de 2 images filtrées par des Gaussiennes de σ différents :

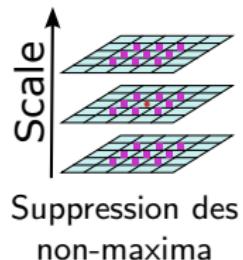
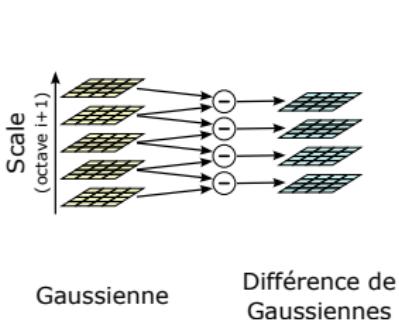
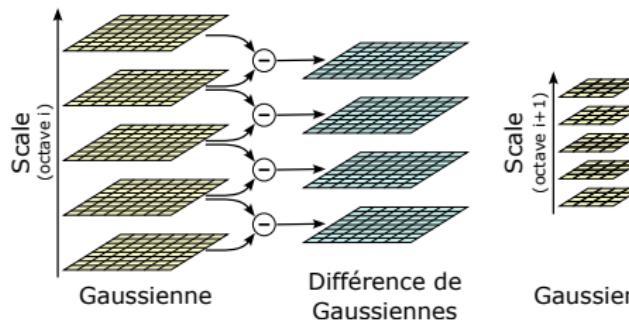
$$G_{k\sigma}(x, y) - G_\sigma(x, y) \approx (k - 1)LoG_\sigma(x, y)$$

- La différence entre 2 images successives (i et $i + 1$) d'une octave permet donc de détecter des blobs de taille σ_i !
- La position des extrema des DoG (dans l'image et dans le log-scale space) permet de détecter les points d'intérêt et leurs échelles.



Remarques

- Il faut faire 1 filtrage de plus que le nombre de DoG souhaitées (puisque il faut faire la différence de 2 images pour avoir 1 DoG)
 - Pour supprimer les non maxima locaux, on compare un point avec ces voisins dans l'image et dans le log-scale space \Rightarrow il faut 2 DoG supplémentaires (une "en-haut" et une "en-bas").
- \Rightarrow Pour pouvoir utiliser n_s scales, il faut $n_s + 3$ images filtrées par des Gaussiennes.



1 Introduction

2 Détection de points d'intérêt

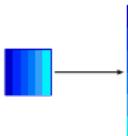
3 Description des points d'intérêt

- Définition
- Contraintes et invariances
- Quelques descripteurs

4 Matching de points d'intérêt

Définition

- Un descripteur capture les informations (image ou autre) dans une région autour d'un point d'intérêt
- L'intensité des pixels autour d'un point d'intérêt (bloc) peut donc être vu comme un descripteur.
- **Remarque :** comme on travaille avec des vecteurs, on empile généralement les colonnes du bloc (conversion matrice → vecteur, cf. cours de Machine Learning de M1)



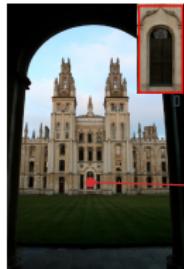
Problème

- Un bloc est un descripteur très limité (pas d'invariances, peu compact, ...)

Contraintes

Pour permettre un matching efficace, le descripteur doit être

- robuste \Rightarrow changement de point de vue 3D, contraste, occlusion partielle
 - discriminant \Rightarrow il faut pouvoir différencier 2 objets (même similaires)
 - compact \Rightarrow il faut parvenir à résumer l'information contenue dans le voisinage
- \Rightarrow un compromis est nécessaire entre la robustesse (nécessite des invariances) et le caractère distinctif (pas trop d'invariances)



Robustesse
(point de vue, occlusion)



Caractère discriminant
(quelle fenêtre choisir ?)

Invariances prises en compte par le détecteur

- Invariance en translation (parcours de toute l'image)
- Invariance en mise à l'échelle : utilisation de l'image dans la pyramide gaussienne correspondant à l'échelle détectée

Invariances à prendre en compte dans le descripteur

- Invariance en rotation
- Invariance à des changements d'intensité

Invariance en rotation

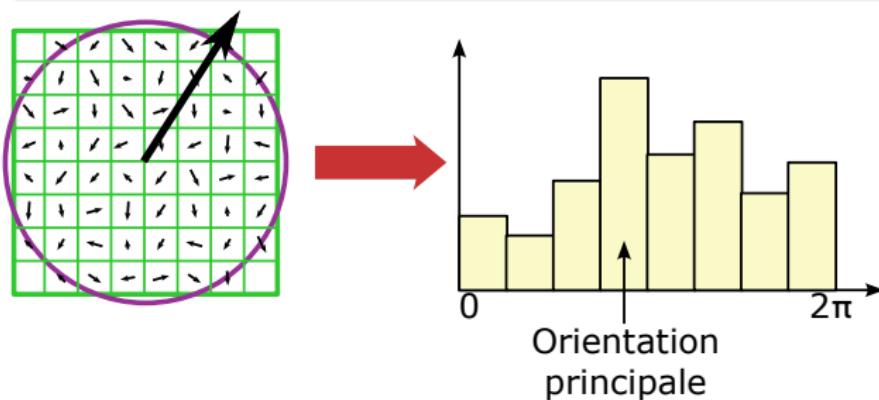
Idée : orientation principale autour du point d'intérêt

- ➊ Calculer l'orientation du gradient pour tous les pixels dans une fenêtre circulaire W autour du point d'intérêt \mathbf{p}_0
- ➋ Calculer l'histogramme des gradients dans cette fenêtre

$$HoG_i(\mathbf{p}_0) = \sum_{\mathbf{p} \in W(\mathbf{p}_0)} \|\nabla I(\mathbf{p})\| \quad \text{si } \theta(\mathbf{p}) \in [ih; (i+1)h[$$

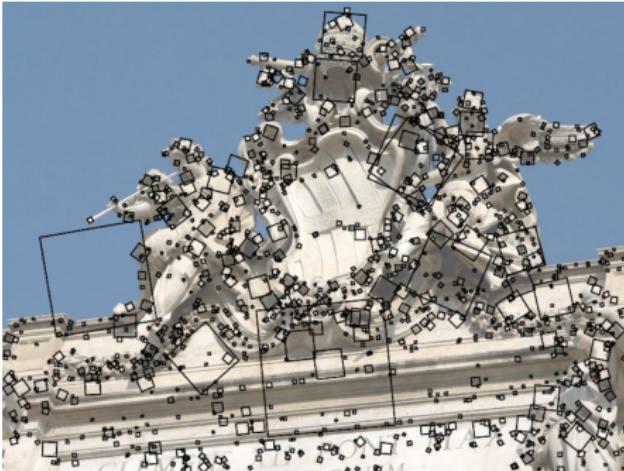
avec $i \in [0; n_b]$, n_b le nombre de bins dans l'histogramme, $h = 2\pi/n_b$ et $\theta(\mathbf{p})$ l'orientation du gradient en \mathbf{p}

- ➌ Extraire l'orientation principale (bin majoritaire)



Idée : orientation principale autour du point d'intérêt

- On utilisera un bloc aligné sur l'orientation principale pour décrire le point d'intérêt
- Si plusieurs orientations ont la même "intensité", on fera plusieurs descripteurs (un par orientation)

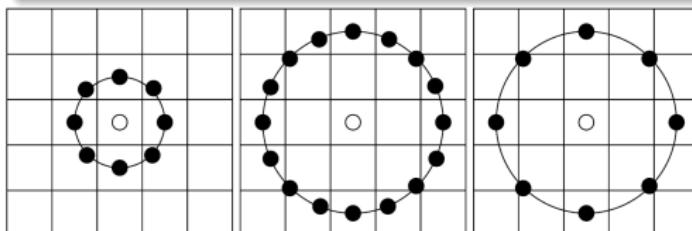


Principe

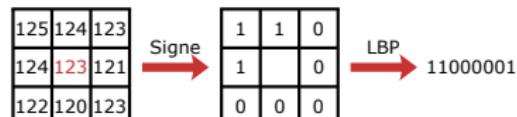
Descripteur simple et rapide à obtenir (équivalent de FAST pour la détection?)

- ➊ On considère n points dans un voisinage circulaire W de rayon r autour du point d'intérêt \mathbf{p}_0
- ➋ Pour chaque voisin \mathbf{p} , on calcule $S(\mathbf{p}) = \begin{cases} 1 & \text{si } I(\mathbf{p}) > I(\mathbf{p}_0) \\ 0 & \text{sinon} \end{cases}$
- ➌ On concatène toutes les valeurs de S pour former un code binaire LBP
- ➍ Pour être invariant en rotation, on peut "tourner" le code binaire de sorte à minimiser sa valeur.

Les codes $10000010b$, $00101000b$ et $00000101b$ correspondent au code minimal $00000101b$



Exemples de voisnages



Process

Principe

Descripteur simple et rapide à obtenir (équivalent de FAST pour la détection?)

- ❶ On considère n points dans un voisinage circulaire W de rayon r autour du point d'intérêt \mathbf{p}_0
- ❷ Pour chaque voisin \mathbf{p} , on calcule $S(\mathbf{p}) = \begin{cases} 1 & \text{si } I(\mathbf{p}) > I(\mathbf{p}_0) \\ 0 & \text{sinon} \end{cases}$
- ❸ On concatène toutes les valeurs de S pour former un code binaire LBP
- ❹ Pour être invariant en rotation, on peut "tourner" le code binaire de sorte à minimiser sa valeur.

Les codes $10000010b$, $00101000b$ et $00000101b$ correspondent au code minimal $00000101b$

Avantages

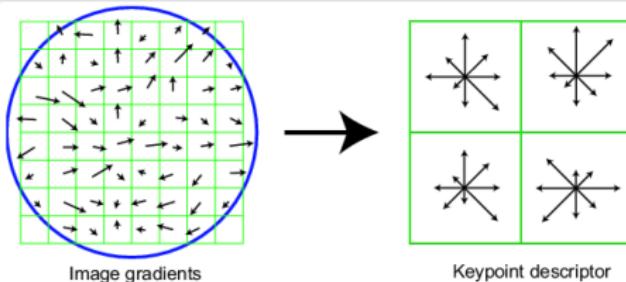
- Rapide à calculer
- Descripteur = un vecteur binaire / un nombre entier \Rightarrow faible dimension et comparaison facile / rapide (distance de Hamming, différence, ...)

SIFT [Lowe 2004] : Histogramme de Gradients (HoG)

Scale Invariant Feature Tracking (SIFT) : Principe

Dans un bloc de 16×16 pixels autour d'un point d'intérêt

- ① Réorienter les gradients de sorte que l'orientation principale du bloc soit de 0° (par exemple)
- ② Diviser le bloc en 4×4 sous-blocs.
- ③ Pour chaque bloc, calculer l'histogramme des gradients (pondérés par leur norme) : utiliser un faible nombre de bins (généralement 8)
- ④ Concaténer les 16 histogrammes \Rightarrow descripteur / vecteur caractéristique de $16 \times 8 = 128$ dimensions
- ⑤ Normaliser le vecteur caractéristique (norme \mathcal{L}_2)



(par simplicité, on représente ici un bloc de 8×8 pixels divisé en 2×2 sous-blocs)

SIFT [Lowe 2004] : Histogramme de Gradients (HoG)

Scale Invariant Feature Tracking (SIFT) : Principe

Dans un bloc de 16×16 pixels autour d'un point d'intérêt

- ① Réorienter les gradients de sorte que l'orientation principale du bloc soit de 0° (par exemple)
- ② Diviser le bloc en 4×4 sous-blocs.
- ③ Pour chaque bloc, calculer l'histogramme des gradients (pondérés par leur norme) : utiliser un faible nombre de bins (généralement 8)
- ④ Concaténer les 16 histogrammes \Rightarrow descripteur / vecteur caractéristique de $16 \times 8 = 128$ dimensions
- ⑤ Normaliser le vecteur caractéristique (norme \mathcal{L}_2)

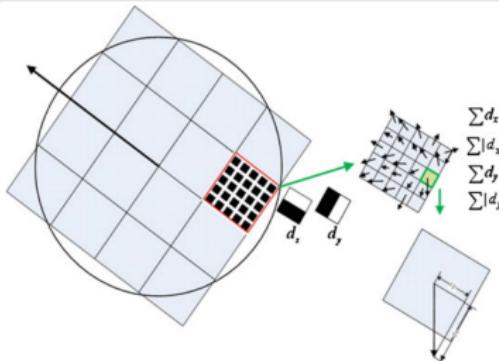
Avantages

- Utilisation du gradient \Rightarrow invariant aux changements d'intensité
- Alignement des orientations sur l'orientation principale \Rightarrow invariant en rotation
- Normalisation du vecteur caractéristique \Rightarrow invariance à des changements d'amplitude du gradient
- Les 16 HoG intègrent une **information discriminative** sur l'orientation

Speeded-Up Robust Feature (SURF) : Principe

Assez similaire à SIFT

- ① Diviser un bloc de 20×20 pixels en 4×4 sous-blocs
- ② Filtrer chaque sous-bloc pour obtenir les dérivées partielles locales en x et y (ondelettes de Haar)
- ③ Réorienter des gradients sur l'orientation majoritaire
- ④ Calculer, pour chaque sous-bloc, $v = (\sum dx, \sum |dx|, \sum dy, \sum |dy|)$
- ⑤ Concaténer v pour chaque bloc \Rightarrow vecteur caractéristique de taille $4 \times 4 \times 4 = 64$ dimensions



Speeded-Up Robust Feature (SURF) : Principe

Assez similaire à SIFT

- ① Diviser un bloc de 20×20 pixels en 4×4 sous-blocs
- ② Filtrer chaque sous-bloc pour obtenir les dérivées partielles locales en x et y (ondelettes de Haar)
- ③ Réorienter des gradients sur l'orientation majoritaire
- ④ Calculer, pour chaque sous-bloc, $v = (\sum dx, \sum |dx|, \sum dy, \sum |dy|)$
- ⑤ Concaténer v pour chaque bloc \Rightarrow vecteur caractéristique de taille $4 \times 4 \times 4 = 64$ dimensions

Avantages

- Utilisation du gradient \Rightarrow invariant aux changements d'intensité
- Alignement des orientations sur l'orientation principale \Rightarrow invariant en rotation
- Utilisation massive d'images intégrales \Rightarrow plus rapide que SIFT

BRIEF [Calonder, Pattern Anal. and Machine Intel., 2012]

Binary Robust Independent Elementary Features (BRIEF) : Principe

Dans un bloc $S \times S$ pixels :

- Tirer "aléatoirement" des couples de points (x_i, y_i) , $i \in [0; N[$ et les comparer

$$\tau_i(x_i, y_i) = \begin{cases} 1 & \text{si } I(x_i) < I(y_i) \\ 0 & \text{sinon} \end{cases}$$

- Réaliser N tests ($N = 128, 256, 512$) et concaténer les résultats sous la forme d'un vecteur binaire

Remarque

- Pour être reproductible, un tirage aléatoire est réalisé une fois pour définir les couples de points.
- Différents types de tirage aléatoire ont été testés par les auteurs.



- 1 Introduction
- 2 Détection de points d'intérêt
- 3 Description des points d'intérêt
- 4 Matching de points d'intérêt
 - Matching
 - Évaluation

Quelques distances utilisables

On considère 2 descripteurs x et y :

- Dans tous les cas :
 - Distance \mathcal{L}_1 ($\sum |x_i - y_i|$), distance \mathcal{L}_2 ($\sqrt{\sum (x_i - y_i)^2}$)
 - Similarité cosinus (= corrélation) :
$$\frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$
- Pour des vecteurs binaires :
 - Distance de Hamming (aka. XOR) : nombre de bits différents entre les 2 codes binaires
 - Indice de Dice, indice de Jaccard
- Pour des histogrammes : distance de Bhattacharyya, entropie croisée

Remarque

- La similarité cosinus (entre 2 blocs d'intensité) peut être biaisée par des changements d'intensité ou de contraste.
- Pour s'affranchir de ce problème, il est préférable d'utiliser la corrélation croisée normalisée

$$\gamma = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

Matching des descripteurs

Principe

Pour chaque point d'intérêt de l'image 1 :

- calculer la distance à tous les points d'intérêt de l'image 2
- considérer comme match / appariement, le point d'intérêt de l'image 2 avec la distance minimale

Problèmes

- Comment gérer des points d'intérêt similaires ?



- Faut-il accepter tous les appariements ? Au delà de quelle valeur de distance, considère t'on que l'appariement est mauvais ?

Comment gérer des points d'intérêt similaires ?

Suppression des mauvais appariements

2 solutions

- Faire un **appariement croisé** : 2 points d'intérêt sont appariés, si et seulement si $d(a_i, b_j) = \min_k(d(a_i, b_k))$ ET $d(b_j, a_i) = \min_k(d(b_j, a_k))$ avec a_k les points d'intérêt détectés dans l'image 1 et b_k ceux de l'image 2
- Calculer le **ratio des distances** entre les 2 meilleurs matchs (distances d_1 et d_2) : si $d_1/d_2 < s$ alors on conserve l'appariement (d_1 est "beaucoup" plus petit que d_2) ; sinon on rejette l'appariement (d_1 est trop proche de d_2).

Remarque

- Rejeter un appariement revient à supprimer le point d'intérêt dans l'image 1

Faut-il accepter tous les appariements ?

Au delà de quelle valeur de distance, considère t'on que l'appariement est mauvais ?

Seuil et courbe ROC

- On peut définir un seuil pour les distances au dessus duquel tout appariement est rejeté
- Choix du seuil empirique ou via une évaluation quantitative (e.g. courbe ROC)

Remarque

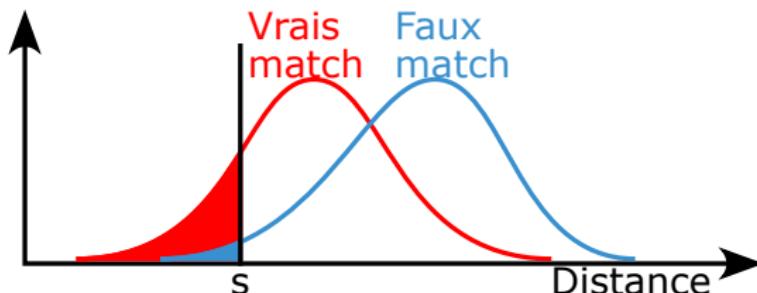
- L'étude des performances du détecteur / descripteur via une courbe ROC permet également de le comparer à d'autres techniques ...

Courbe ROC : quelques définitions / rappels

Pour une valeur de seuil donnée

- Taux de vrais positifs (TPR) : $\tau_+ = \frac{N_+}{N_t}$
- Taux de faux positifs (FPR) : $\tau_- = \frac{N_-}{N_t}$

avec N_t le nombre total de matchs, N_+ le nombre de matchs corrects, N_- le nombre de mauvais matchs



Courbe ROC : construction

- Faire varier la valeur de seuil de sorte à considérer toutes les valeurs possibles (et conduisant à des changements du TPR ou du FPR)
- Placer chaque couple (TPR, FPR) sur une graphe
- Calculer l'aire sous la courbe (AUC)

Courbe ROC : analyse

- $AUC = 0.5 \Leftrightarrow$ mauvais descripteur (assignation aléatoire des appariements)
- $AUC = 1 \Leftrightarrow$ bon descripteur (appariements parfaits, on ne se trompe jamais)
- Le seuil idéal / point de fonctionnement correspond au point de la courbe ROC le plus proche du point $(0, 1)$ (indice de Youden)

