Rendu TP 3 : SIFT

EL OUAFI, Moussa

30 mars 2021

Ce TP porte sur l'algorithme SIFT. Les documents concernés sont :

1. L'article IPOL "Anatomy of the SIFT method"
   http://www.ipol.im/pub/art/2014/82/
2. Les articles et codes originaux de David Lowe
   http://www.cs.ubc.ca/~lowe/keypoints/

SIFT en deux pages

Voici une description haut niveau de l'algorithme SIFT.

La méthode SIFT sert à trouver des points correspondants entre deux images. Suivant un schéma habituel en traitement d'image, la correspondance se déroule en deux étapes. La première étape est l'extraction de caractéristiques *(feature extraction)*, qui traite chaque image indépendamment, afin d'extraire de chacune un ensemble de descripteurs. La deuxième étape est la mise en correspondance de ces descripteurs *(feature matching)* qui produit un ensemble de paires de points entre les deux images

Algorithm 1: `sift-feature-extraction`

Input : Image $A : \Omega \to \mathbf{R}$
Output: Descripteurs : $\mathcal{S}(A) = \{(x_i, \sigma_i, \theta_i, d_i)\} \subseteq \Omega \times \mathbf{R}^+ \times [0, 2\pi) \times \mathbf{R}^d$

// compute gaussian scale space and D.O.G.
1 $G(x, \sigma_j) \leftarrow G_{\sigma_j} * A(x) \qquad x \in \Omega,\ \sigma_j = \sigma_0, \ldots, \sigma_{max}$
2 $L(x, \sigma_j) \leftarrow G(x, \sigma_j) - G(x, \sigma_{j-1}) \qquad x \in \Omega,\ \sigma_j = \sigma_1, \ldots, \sigma_{max}$
// find keypoint positions
3 $\{(x_i, \sigma_i)\}_{i=1,\ldots N} \leftarrow \operatorname{loc\,max}_{x,\sigma} |L(x, \sigma)|$
4 refine positions x_i to sub-pixel precision, filter and remove keypoints on edges
// find keypoint orientations
5 $\theta_i \leftarrow$ dominating orientation of $\nabla_x G(x_i, \sigma_i)$ in a neighborhood of x_i of size σ_i.
// compute keypoint descriptors
6 **for** $i \in 1 \ldots N$ **do**
7     $d_i \leftarrow \vec{0}$
8     $I \leftarrow 16 \times 16$ samples of $\nabla_x G(x, \sigma)$ around (x_i, σ_i) (rotated by θ_i)
9   **for** $p, q \in 0 \ldots 3$ **do**
10         $J \leftarrow 4 \times 4$ sub-image of I indexed by (p, q)
11         $H \leftarrow$ smoothed, weighted and filtered histogram of atan2(J), using 8 bins
12         $k \leftarrow 4p + q$
13         $d_i(8k, \ldots, 8k+7) \leftarrow H$
14 **return** $\{(x_i, \sigma_i, \theta_i, d_i)\}$

Commentaires.

A1, lignes 1,2 : les images discrètes $G(\cdot, \sigma)$ et $L(\cdot, \sigma)$ sont de plus en plus lisses quand σ augmente, donc on peut des échantillonner de façon de plus en plus grossière.

A1, ligne 3 : "loc max" indique l'ensemble de maxima locaux d'une fonction 3D

A1, ligne 5 : l'orientation dominante se définit comme la mode de l'histogramme d'angles du champ de vecteurs $\nabla_x G(x_i, \sigma_i)$.

A1, ligne 8 : les angles de I sont pris par rapport à l'orientation dominante

A1, lignes 4,11 : voir l'article pour les détails de ces pondérations, interpolations et seuillages

Algorithm 2: `sift-feature-matching`

Input : Image $A : \Omega \to \mathbf{R}$
Input : Image $A' : \Omega' \to \mathbf{R}$
Output: Matches $(x, x') \subseteq \Omega \times \Omega'$

// compute sift descriptors of each image
1 $S \leftarrow$ `sift-feature-extraction`(A)
2 $S' \leftarrow$ `sift-feature-extraction`(A')
// find correspondences between descriptors
3 $R \leftarrow \emptyset$
4 **for** $(x, \sigma, \theta, d) \in S$ **do**
5   **for** $(x', \sigma', \theta', d') \in S'$ **do**
6   **if** *distance*$(d, d') <$ *small enough* **then**
7             $R \leftarrow R \cup \{(x, x')\}$
8 **return** R

A2, ligne 8 : un critère simple pour "*distance*$(d, d') <$ *small enough*" est fixer un seuil absolu, par exemple *distance*$(d, d') < 300$. Un critère un peu plus robuste consiste à fixer un seuil relatif : on garde seulement le d' qui minimise la distance avec d, mais seulement si cette distance est plus petite que 0.8 fois la distance avec d'', le deuxième descripteur plus proche de d. Habituellement on utilise ce critère plus robuste. Le facteur 0.8 a été choisi de façon expérimentale.

0. Questions générales sur SIFT

0.1. Quel est le problème que la méthode SIFT vise à résoudre ?

La méthode SIFT cherche à transformer une image en vecteurs caractéristéques invariants par les transformations géométriques. Parmis les transformations gémetrique on cite la rotation, la translation et la mise à l'echelle.

0.2. Les "features" extraites sont invariantes par rotation. Quelle étape de l'algorithme assure cette propriété ?

Pour que les "features" extraites soient invariantes par rotation, l'algorithme SIFT Calcul des histogrammes d'orientations en fonction du voisinage des points. l'etape concerné est donc :

// find keypoint orientations

$\theta_i \leftarrow$ dominating orientation of $\nabla_x G(x_i, \sigma_i)$ in a neighborhood of x_i of size σ_i.

Cette étape est nécessaire pour assurer l'invariance par rotations des discripteur calculés dans l'étape *// compute keypoint descriptors*

.

$\star$**0.3.** La méthode n'est pas invariante par changement de contraste linéaire. Pourquoi ? Suggérez un minimum de modifications pour que la méthode soit invariante.

Le changement de contraste linéaire détruit l'invariation par mise en échelle comme dans le cas de SCB, la solution est donc d'appliquer changement de contraste linéaire avant LE SIFT.

1. Questions sur l'espace d'échelle Gaussien

Dans cette question on travaille dans l'onglet de la demo nommé *"Examine Gaussian scale-space computation"*. Notamment, la gallerie d'images qui montre l'espace d'échelle des images et de leurs laplaciens.

$\star$**1.1.** La différence de Gaussiennes introduit une troisième méthode de calcul du Laplacien d'une image discrète. Veuillez rappeler les deux autres méthodes que l'on a déjà vu dans ce cours, avec une notation commune (en précisant exactement comment calculer le laplacien d'une même image image $I(i,j)$ dans chacun des trois cas).

- **La différence de Gaussiennes** :
  on fait la convolution d'image I par une gaussienne G_σ, on pose

  $w(\sigma,x,y) = (G_\sigma * I)(x,y)$. on a $\Delta G_\sigma * I = (\Delta G_\sigma) * I$. donc $w(k\sigma,x,y) - w(\sigma,x,y) \approx (k-1)\sigma^2(\Delta G_\sigma * I)(x,y)$. par l"équation de chaleur.

- **Différence finie** : $\Delta_d I(i,j) = I(i+1,j) + I(i-1,j) + I(i,j+1) + I(i,j-1) - 4I(i,j)$.
- **Transformation de Fourier Discrète** :
  On utilise la méthode vu au section "5.3 Poisson Image Editing Equations with the Fourier Method" de cours.
  La TFD est un isomorphisme donc on calculant la tranformation de Fourier de laplacien de l'image I ie la fonction $\delta_d I : (i,j) to I(i+1,j) + I(i-1,j) + I(i,j+1) + I(i,j-1) - 4I(i,j)$ on peut déduire le laplacien discrèt par application de TFD inverse. Par le caractère Shift de TFD on trouve que pour une image I de taille $J \times L$:

  $(\mathcal{F}(\Delta_d I))(m,n) = ((\frac{2\pi m}{J})^2 + (\frac{2\pi n}{L})^2) = (\mathcal{F}(I))(m,n)$
  par suite $\Delta_d I(i,j) = \mathcal{F}^{-1}(((\frac{2\pi i}{J})^2 + (\frac{2\pi j}{L})^2) I(i,j))$.
  donc il faut calculer la TFD inverse de la fonction $(i,j) \to ((\frac{2\pi i}{J})^2 + (\frac{2\pi j}{L})^2) I(i,j)$ ce qui est facile à faire.

1.2. Dans la gallerie, on voit les laplaciens normalisés affichés avec un code de couleur (bleu-blanc-rouge). Quel est le signe qui correspond à chaque couleur ? (positif, négatif, zéro).

- Bleu $\longleftrightarrow$ positif.

- Blanc $\longleftrightarrow$ zéro.

- Rouge $\longleftrightarrow$ négatif.

1.3. Pourquoi, sur les dernières échelles de la pyramide, y observe-t-on des grands carrés de la même couleur ?

la pyramide permet de représenter le contenu d'image en niveaux de gris en combinant des opérations de sous-échantillonnage avec une étape de lissage. Passer d'une octave à la suivante on double le paramètre σ. L'image utilisée pour créer l'octave suivante est donc l'image de param-tre 2σ et dont les dimensions sont divisée par deux. Sur la base de la pyramide se trouve l'image originale et à l'échelle suivante, la rosolution est divisé par 2 donc on obtient dans la dernière échelles une image flou donc des grandes carrée de meme couleur qui caractérise leur niveaux de gris.

1.4. Quel est l'intérêt d'utiliser un espace d'échelle, pour l'objectif final ?

l'intérêt d'utiliser un espace d'échelle est d'étudier des propriétés de l'image qui ne peuvent pas mise en évidence qu'à un échelle bien précis. l'espace d'echelle nous permet donc de décrire le contenu d'image sur différentes écheclles.

Pour ce faire l'algorithme SIFT, fait la représentation d'image sur plusieurs échelles utilisé est une pyramide de gaussienne.

2. Questions sur la détection de *keypoints*

Dans cette question on travaille dans l'onglet de la demo nommé *"Examine keypoints detection"*. Notamment, sur la gallerie avec les images $A, B, C, \ldots$

2.1. Veuillez expliquer comment on calcule les points de l'image A à partir de l'espace d'échelle Gaussien construit en l'étape antérieure.

Après qu'on a procédé par DoG (Algorithme 3) entre deux images consécutives d'une memê octave dans la pyramide de gaussiennes pour obtenir une pyramide de DoG. $\mathcal{D}(x,y,\sigma) = w(k\sigma,x,y) - w(\sigma,x,y)$ avec k nombre constant qui assure l'obtention d'un nombre fixe d'images lissées par octave, il garanti que on a le meme nombre de DoG par octave.

Algorithm 4 : Scanning for 3d discrete extrema of the DoG scale-space : consiste de trouver Les points d'intérêts recherchés (A) constituent les extrema locaux des images des DoG à travers les différentes échelles. Chaque pixel des images des DoG est alors comparé à ses (26 = 3 × 3 × 3 - 1) voisins si le point est un extema local on le garde.

les points A assure l'invariance par mise en échelle.

$\star$**2.2.** Veuillez expliquer comment calculer les points B à partir des points A.

Algorithm 5 : Discarding low contrasted candidate keypoints (conservative test) consiste à rejeter des points A instables et on garde que les points extremas qui verifient :

$$w^o_{s,m,n} \geq 0.8 C_{DoG}$$

cette étape permet de rejeter les extremas à faible contraste, ceci assure l'invariance des Points d'intérêts à l'illumination.

$\star$**2.3.** Veuillez expliquer comment calculer les points C à partir des points B.

cette question concerne **Algorithm 6 : Keypoints interpolation**. Pour raffiner la position d'un extrema discret (s_e, m_e, n_e) dans un octave o_e.

- On prend (s, m, n) ses cordonnées discrètes
- on calcule l'extrema continue de la fonction $w^o_{s,m,n}$ en résolant $\nabla w^o_{s,m,n}(\alpha) = 0$. on obtient

$$\alpha^* = -(\bar{H}^o_{s,m,n})^{-1}(\bar{g}^o_{s,m,n})^{-1}.$$

- si $max(|\alpha_1|, |\alpha_2|, |\alpha_3|) \leq 0.6$ l'ectrema est donc accepté, ses corrdonées sont données par la formule (15)

- si $max(|\alpha_1|, |\alpha_2|, |\alpha_3|) > 0.6$ on rejette cet extrema et on prend comme nouvelles cordonnées $(s, m, n) + \alpha^*$ et on répete le test pour cette nouvelle coordonnés. si après 5 itérations ces nouvelles coordonées ne se trouvent pas dans le domaine de validite

- On prend (s, m, n) ses cordonnées discrètes on rejet l extrema discret (s_e, m_e, n_e).

ce processus est fait pour tous les extremas, on garde parsuite que les points validées par l'alorithme si dessus.

remarque : le domaine de validitée est caractérisé par $max(|\alpha_1|,|\alpha_2|,|\alpha_3|) \leq 0.5$

cette étape nous permet de trouver la valeur DoG corréspondante des points C.

$w = w^o_{s,m,n} + \frac{1}{2}(\alpha^*)^T(\bar{g}^o_{s,m,n})^{-1}$. utilsé en algorithme 5.

$\star$**2.4.** Veuillez expliquer comment calculer les points D à partir des points C.

consiste à rejeter des points C instables et on garde que les points extremas qui verifient :

$$w^o_{s,m,n} \geq 0.8C_{DoG}$$

Remarque : cette étape n'est pas une répition d'étape faite en (2.2). en effet les nouvelles valeurs $w^o_{s,m,n}$ calculer peuvent etre inférieur $0.8C_{DoG}$ donc ils ont des contraste faible, une chose qui peut créer des éctremas faux. Donc on les rejette.

$\star$**2.5.** Veuillez expliquer comment calculer les points E à partir des points D.

cette étape consiste à éliminer les points d'interes sur les bords qui sont difficile à localiser. Pour ce faire on calcule la matrice Hessiene de DoG dans les points extremas.

Le SIFT rejete parsuite les points dont $r = \frac{\lambda_{max}}{\lambda_{min}} > C_{edge}$ (on prend $C_{edge} = 10$ par défaut).

puisque le calcul des valeurs propres $\lambda_{max}, \lambda_{min}$ peut s'averer difficile. donc une méthode alténative est de calculer

$edgeness(H^o_{s,m,n}) = \frac{r^2+1}{r}$ et éliminer les points ayant $edgeness(H^o_{s,m,n}) > \frac{C^2_{edge}+1}{C_{edge}}$.

2.6. Quelle nouvelle information y-a-t'il dans les points "SIFT" par rapport aux points "E" ? Comment cette information est-elle calculée ?

La nouvelle information est **l'orientation de référence**.

Il s'agit à présent d'attribuer à chaque Point d'Interet sélectionné une ou des orientations en utilisant la direction des gradients des voisins directes de ce point. Pour cela on parcourt tous les pixels de toutes les images gaussiennes à toutes les octaves et on leur affecte une orientation et une norme.

2.7. Qu'indique le rayon des cercles autour des points ?

le rayon des cercles corréspond au l'écart type de la gaussienne (Standard Deviation) ou la taille d'une fenêtre gaussienne.

— Cercle bleu : rayon = σle facteur d'échelle σ.
— Cercle vert :rayon =$\lambda_{ori}\sigma$, avec λ_{ori} est un paramètre assigné par l'utilisateur, il est utile pour préciser l'orientation des régions formées des pixels qui contibuent à l'orientation dans un vosinage précis.
— Cercle rouge : rayon =$\lambda_{descr}\sigma$, avec λ_{descr} est un paramètre assigné par l'utilisateur qui sert à réduire la contribution des pixels distants, et de calculer l'orientation des gradiants des régions de patch carré normalisé.

2.8. Quel est l'intérêt de définir l'orientation d'un point ?

l'intérêt est pour fixer une orientation de référence, et assurer l'invariance par rotation de descripteur SIFT.

3. Questions sur les descripteurs SIFT

Dans cette question on travaille dans l'onglet de la demo nommé *"Examine keypoints description and matching"*. Concrètement, sur la grille de 4×4 histogrammes qui apparaissent quand on clicke sur un des points des images.

3.1. Décrivez ce que sont ces 4×4 histogrammes par rapport au morceau d'image que l'on voit à droite.
on rappelle que les coleurs rouge, bleu e

Les histogrammes contient les orientations (gradiant oriantations) de chaque pixel du patch carré $\mathcal{P}^{descr}$.

3.2. Ces 16 histogrammes sont calculés à partir de l'imagette à droite. Décrivez comment seraient chacun de ces 16 histogrammes si l'imagette était un disque noir sur fond blanc, de même diamètre que l'imagette.

Si on change l'imagette était un dique on perdera certainement certains pixeels qui contribuent à l'orientation. Ceci dit, on aura un nombre inférieur de bins avec valeur différent en norme et direction de gradients.

3.3. Même question que (3.2), mais l'imagette étant le coin d'un carré noir sur fond blanc, avec le coin supérieur gauche du carré juste en dessous à droite du centre de l'imagette.

on aura une orientation des gradients au sense inverse.

3.4. Le descripteur SIFT, comme objet, est un vecteur de l'espace $\mathbf{R}^d$. Que vaut d ? Quel est le rapport de ce vecteur avec les 4×4 histogrammes ?

$d = n_{hist} \times n_{hist} \times n_{ori}$. avec, n_{hist} de sorte que $h^{i,j}$ les histogramme qui forme le vecteur $n_{hist} \times n_{hist} \times n_{ori}$. d'orientation du gradiant sont associéu à une position "keypoint" $(x_{key}, y_{key}, \sigma_{key}, \theta_{key})$. telle que chaque $h^{i,j}$ est formé de n_{ori} barres d'histogramme (histogramme bins).

dans le cas ou $n_{hist} = 4$ (la valeur par défaur d'algorithme SIFT) ce vecteur est égale à 4×4 histogrammes vu en tant que vecteur.

4. Questions sur la mise en correspondance de deux images

$\star$**4.1.** Si vous avez deux descripteurs SIFT, comment pouvez-vous calculer la distance entre eux ?

On va utiliser la distance euclidienne usuelle dans le plan d_2, pour deux point $a = (x_a, y_a)$, $b = (x_a, y_b)$ du plan euclidien de dimension 2. definie par $d_2(a,b)^2 = (x_a - b_x)^2 - (y_a - y_x)^2$ pour creér deux vecteurs lignes "distance" qu'on nome $D1$

et $D2$ de taille $N_1 \times N_2$ comme suit :

soient N_1, N_2 réspictivement le nombre de keypoints du $SIFT1$ et du $SIFT1$.
de taille $M = max(N_1, N_2)$

- Pour $1 \leq i \leq N_1$ on crée un vecteur ligne $DIST1(i)$ de taille N_2 telle que :

$1 \leq \forall j \leq N_2$, $DIST1(i)(j) = d_2(SIFT1(i), SIFT2(j))$

- Pour $1 \leq j \leq N_2$ on crée un vecteur ligne $DIST2(i)$ de taille N_2 telle que :

$1 \leq \forall i \leq N_1$, $DIST2(i)(j) = d_2(SIFT1(i), SIFT2(j))$

on pose $D1 = (DIST1(1), \cdots, DIST1(N_1))$ et $D2 = (DIST2(1), \cdots, DIST2(N_2))$.

on définit donc la distance d_{SIFT} comme suit :

$d_{SIFT}(SIFT1, SIFT2) = (\sum_{i=1}^{N_1 \times N_2} (D1(i) - D2(i))^2)^{\frac{1}{2}}$.

cette distance est bien une distance euclidienne. qui définie bien "nearest and second nearest neighbod".

4.2. La méthode de correspondance SIFT sert à choisir des paires de points provenant de deux images quand leurs descripteurs respectifs sont suffisamment proches. Le critère habituel est un seuillage relatif, par rapport à la distance du deuxième descripteur plus proche. Pourquoi ce critère est-il raisonnable ?

On utilise le seuillage relatif pour éviter la dépendance du descripteur de la valeur absolue ; une chose que peut résulter de changer certaines invariance par transformations géomitrique que le SIFT cherche à assurer. En particulier la rotation.

4.3. Décrivez une paire d'images particulière où le critère de seuillage relatif n'est pas adapté, et il rejetterait la plupart de bonnes correspondances.

une image flou.