

TP Transformée de Hough

(C. Burnier – Jean-Pierre Bruandet – Xavier TROUILLOT)

Dans ce TP, on se propose d'implémenter la transformée de Hough pour la détection de points alignés. Dans un deuxième temps, on s'intéressera à la transformée de Hough aléatoire (Randomized Hough Transform), qui est un des moyens d'accélérer le processus.

1. Transformée de Hough pour la détection de droite

Initialisation

1. Charger le fichier matlab et l'image 'maison.jpg'
2. A quoi correspondent les variables : « `ro_max`, `ro_min`, `theta_max`, `theta_min`, `dro`, `dtheta` » ? Cela vous semble-t-il approprié ? Justifier.
3. Initialiser à zéro le tableau `Hough` qui représentera l'espace de Hough.
4. Sélectionner les indices `ind` correspondant aux pixels non nul de l'image (commande `find`) et les convertir en indice ligne/colonne (`i,j`) (commande `ind2sub`).

Transformée de Hough

5. On choisit comme repère les axes (`x,y`) avec `x`, coordonnées des colonnes et `y`, opposée de la coordonnées des lignes. Ecrire (en commentaire ou sur une feuille) la relation liant `x` à `j`, `y` à `i`. Dans une boucle, portant sur tous les pixels de `ind`, convertir (`i,j`) en (`x,y`).
6. Construire une fonction « `Hough=SinusHoughSpace(Hough, x,y,ro_max)` » qui pour un point (`x,y`) donné, trace la sinusoïde correspondante. On n'oubliera pas que les valeurs de `Hough` doivent être accumulées au fur et à mesure des appels à la fonction « `SinusHoughSpace` ».

NB : on pourra prendre comme convention qu'un pixel (`l,k`) de l'espace de Hough représente les valeurs de (`ro,theta`) en $ro=ro_min+dro/2 + (l-1)dro$ et $theta=theta_min+theta/2 + (k-1)dtheta$. Se représenter cet échantillonnage.

7. En dehors de la boucle, afficher l'image correspondant à l'espace de Hough.

Tracé des droites correspondant aux premières valeurs maximales de l'espace de Hough

8. Dans un second temps, on recherche les 30 premières valeurs maximales de l'espace de Hough et on trace les droites correspondantes. Pour cela :
 - a. Dans une boucle itérant sur les 30 premières valeurs maximales de l'espace de Hough, rechercher l'(es) indice(s) (l,k) correspondant à la plus grande valeur de l'espace de Hough.
 - b. Ecrire une fonction `[ro,theta]=Hough2ImSpace(l,k,dro,dtheta,ro_min,theta_min)` qui transforme les indices (l,k) de l'espace de Hough en leurs coordonnées réelles (ro,theta). Ce couple correspond à la droite cherchée dans l'espace image.
 - c. Mettre à zéros la valeur dans l'espace de Hough correspondant à (l,k). Pourquoi ?
 - d. Ecrire une fonction `[P1,P2]=DeuxPoints(ro,theta,H,W)` qui pour un couple (ro,theta) renvoie deux points extrêmes de l'image (par exemple P1(?,1) et P2(?,W)) appartenant à la droite définie par (ro, theta). P1 et P2 sont données en coordonnées pixels.
 - e. Afficher alors sur l'image de départ la droite passant par P1 et P2 :
`figure(1), hold on, plot([P1(1) P2(1)], [P1(2) P2(2)], 'r');`
9. Que constatez-vous ?

Normalisation

10. Avant la partie « tracé des droites » et juste après le calcul de la transformée de Hough, si le booléen `normalisation` est vrai (on initialisera ce booléen en tête du programme), alors calculer la transformée de Hough d'une image qui vaudrait 1 partout. Bien sûr, il est inutile de créer cette image.
11. Normaliser alors la transformée de Hough de l'image « maison.jpg » par celle de l'image à 1.
12. Exécuter à nouveau le programme jusqu'à sélection des droites.
13. Que représente cette normalisation ? Analyser les résultats.

Pour aller plus loin (facultatif)

Les deux améliorations possibles étant indépendantes, on pourra s'atteler à l'une ou/et l'autre.

1. Sélection des droites dans l'espace de Hough

Une variante de la méthode précédente est de sélectionner, dans l'espace de Hough, un autre « pic » que les n plus grandes valeurs l'espace de Hough. On pourra choisir :

- Les N plus grands maxima locaux de la transformée de Hough
- Les barycentres autour des n premières plus grandes valeurs de l'espace de Hough
càd : si $p=(k,l)=$ un des premiers pics de Hough et $v=$ voisinage autour de p
alors $b = \sum_{v_i \in v} Hough(v_i) v_i$

2. Randomized Radon Transform

Une astuce pour rendre plus rapide le calcul de la transformée de Hough (surtout lorsqu'il s'agit de déterminer des courbes qui doivent être plus de trois de paramètres) est de considérer des n -uplets de points (couples, triplets...de points) dont la dimension permet de déterminer de manière unique le type de courbe cherchée. Par exemple, une droite peut être déterminée de manière unique par un couple de points.

- On tire aléatoirement des couples de points (dont les pixels sont à 1).
- On détermine les paramètres (ro, theta) de la droite contenant les deux points.
- On incrémente dans l'espace de Hough le point (ro,theta).
- On réitère la procédure jusqu'à ce que le maximum dans l'espace de Hough est atteint une certaine valeur seuil.
- On trace alors la droite correspondant au maximum dans l'espace de Hough.

Algorithme

Hough_maxi=init() ;

Pour d=1 : nombre de droites à identifier

```

Hough=zeros (N_ro,N_theta)
TantQue max(Hough) <> Hough_maxi
    P1=random(Un point de l==1)
    P2= random(Un point de l==1)
    (ro,theta)=Paramètres Droite(P1,P2)
    (l,k)=conversionpixelunit(ro,theta) ;
    Hough(l,k)=Hough(l,k)+1
FinTantQue
(l,k)=argument Maximum (Hough)
(ro,theta)=conversion(l,k)
TacerDroite(ro,theta)

```

FinPour

Implémenter la méthode Randomized Radon Transform. On pourra chercher à identifier 10 droites. On pourra prendre Hough_maxi=10.