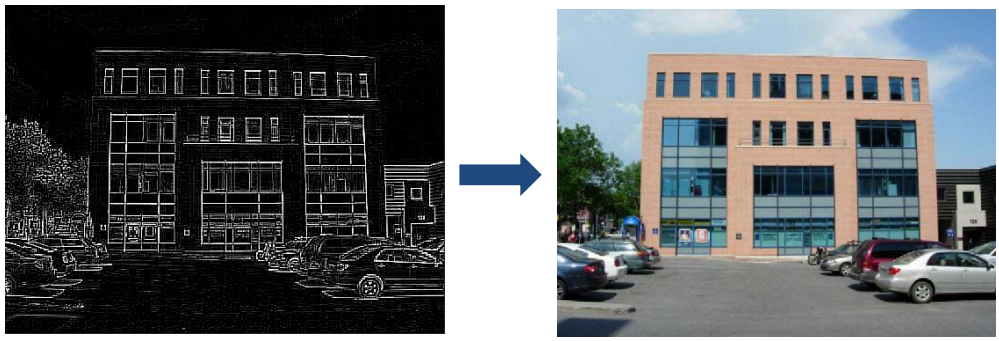
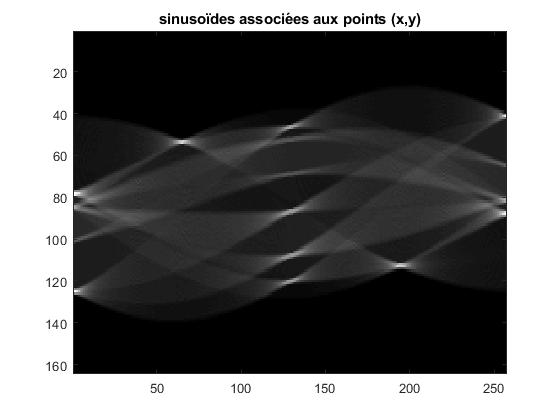
**Transformée de Hough**





**Sommaire**

Introduction

Initialisation

Transformée de Hough

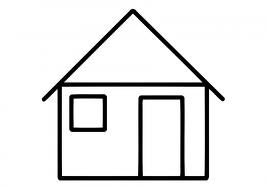
Tracé des droites correspondant aux premières valeurs maximales de l’espace de Hough

Normalisation

Conclusion

**Introduction**

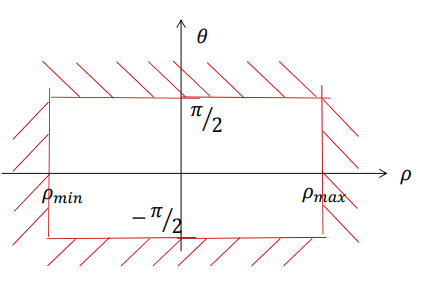
L’objectif de ce TP est d’implémenter la transformation de Hough pour la détection de points alignés. On s’intéressera dans notre cas à une image simple d’une maison ; puis on implémentera la transformée de Hough à cette image avant de tracer les droites correspondant aux premières valeurs maximales de l’espace de Hough.



Rappelons tout d’abord le principe de la transformation de Hough. On considère l’équation d’une droite dans un plan avec deux paramètres (a,b) comme étant : . Ainsi, toute droite traversant notre image binaire peut être mise sous cette forme, et peut être représentée par un point dans le plan (a,b). Si on considère M(x,y) un point quelconque de l’image binaire, l’ensemble des droites passant par ce point est représenté par une droite dans le plan (a,b). La méthode de Hough consiste alors à utiliser un accumulateur dans le plan (a,b). Pour chaque pixel M(x,y) de l’image, la droite est tracée sur l’accumulateur : plus précisément, chaque pixel rencontré est incrémenté d’une unité. Finalement, les éléments de l’accumulateur les plus peuplés correspondent à des droites détectées. Afin de couvrir à la fois les petites, les moyennes et les grandes valeurs des coefficients (a,b), et comme les valeurs possibles de a et de b ne sont pas bornées, on utilise les coordonnées polaires : .

**Initialisation**

En utilisant des coordonnées polaires, notre nouvel espace d’image est dorénavant borné. En effet, on a , et .



Ainsi, les variables *ro\_max*, *ro\_min*, *theta\_max* et *theta\_min* de notre code représente cette espace bornée. La variable *dro* représente la largeur de détection des valeurs maximales de l’espace de Hough. Plus cette variable sera petite, moins on pourra détecter les valeurs maximales associées à la sinusoïde des points (x,y). Et inversement, plus la valeur de cette variable est grande, plus on détecte des valeurs maximales au détriment de la qualité de la sinusoïde associée au point (x,y). Il faut ainsi trouver une bonne valeur de *dro* afin de détecter les valeurs maximales sans perdre la qualité de la sinusoïde ; *dro*=5 semble convenir dans notre situation.

Quant à la variable *dtheta*, celle-ci représente l’angle associée aux droites par rapport au pixel inférieur gauche de l’espace.

Comme nous avons une image binaire, nous cherchons à chercher les indices des pixels non nuls de l’image. Pour ce faire, on utilise la commande find qui permet de trouver les pixels non nuls de l’image, puis nous utilisons la commande ind2sub pour les convertir en indice ligne/colonne (i,j).

|  |
| --- |
| [ind]=find(I); %indice des pixels non nuls de l'image  [i,j] = ind2sub([H,W],ind); % conversion en indice ligne/colonne |

**Transformée de Hough**

En choisissant un repère d’axe (x,y), avec x la coordonnée des colonnes, et y l’opposée de la coordonnée des lignes, on peut convertir les indices lignes/colonnes (i,j) en (x,y) à l’aide des relations suivantes :

Ce nouveau repère permet pour un point (x,y) donné, de tracer la sinusoïde correspondant. Afin de respecter les consignes de l’énoncé, nous avons dû prendre comme convention qu’un pixel (l,k) de l’espace de Hough représente les valeurs de (rho, theta) en et .

Pour réaliser ceci, nous avons codé une fonction SinusHoughSpace qui prends comme valeur d’entrée le tableau Hough, les variables x et y, ainsi que ro\_max.

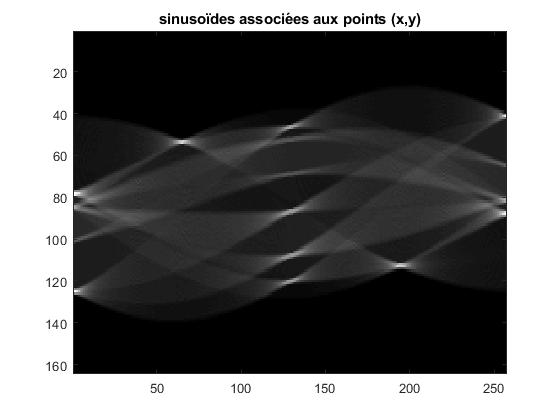
Lorsque tous les points de l’image ont été traités, on sélectionne les éléments de l’accumulateur remplis au-dessus d’un certain seuil, qui sont alors interprétés comme les paramètres des droites correspondant à des lignes droites sur l’image. Le nombre d’éléments de l’accumulateur doit être choisi en fonction de la précision souhaitée. Cette précision a bien sûr un coût : plus le nombre de points est élevé, plus la sinusoïde doit être échantillonnée finement. D’où l’intérêt de choisir de bonnes valeurs de variable dro et dtheta.

|  |
| --- |
| Hough = zeros(Nro,Ntheta); %initialisation du tableau de Hough à zero  x0=W/2;  y0=H/2;  for n=1:length(ind)  %relation liant x à j et y à i permettant la conversion de (i,j) en  %(x,y)  x= j(n) - x0;  y= -(i(n)-y0);    %fonction retournant la sinusoïde correspondant au point (x,y)  Hough=SinusHoughSpace(Hough,x,y,ro\_max);  end  figure(2);  colormap gray;  imagesc(Hough);  title('sinusoïdes associées aux points (x,y)'); |

Fonction SinusHoughSpace :

|  |
| --- |
| function Hough=SinusHoughSpace(Hough,x,y,ro\_max)    %variables nécessaires au calcul de la sinusoïde  dro=10; %largeur trait  dtheta=atan(dro/ro\_max); %angle des droites (centre, pixels coin inf-gauche)  theta\_max=pi/2;  theta\_min=-pi/2;  Ntheta=round((theta\_max-theta\_min)/dtheta);  dtheta=(theta\_max-theta\_min)/Ntheta;  ro\_min=-ro\_max;  Nro=round((ro\_max-ro\_min)/dro);  dro=(ro\_max-ro\_min)/Nro;  [H,W]=size(Hough); %récupération de la taille du tableau Hough    for theta=theta\_min:dtheta:theta\_max  ro=x\*cos(theta)+y\*sin(theta); %on trouve ro pour un couple (x,y)  l=ceil(1+(ro-ro\_min-dro/2)/dro); %on trouve l avec les formules du NB du sujet  k=ceil(1+(theta-theta\_min-dtheta/2)/dtheta);%on trouve k avec les formules du NB du sujet  if (l<=H & k<=W)  Hough(l,k)=Hough(l,k)+1; %on trace la sinusoide  end  end  end |

Finalement, on obtient la figure suivante :



On constate bien sur cette figure des points d’accumulation de droites. On cherchera par la suite à extraire ces points-là pour tracer les droites correspondant aux premières valeurs maximales de l’espace de Hough.

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

Dans cette partie, nous allons nous focaliser seulement sur les 30 premières valeurs maximales de l’espace de Hough afin de tracer les droites correspondantes.

Pour ce faire, nous avons réalisé une boucle avec 30 itérations afin de chercher les indices (l,k) de ces plus grandes valeurs.

Le code correspondant est le suivant :

|  |
| --- |
| %recherche des 30 premières valeurs maximales de l'espace de Hough  Hough\_s = zeros(Nro,Ntheta); %Initialisation d'un tableau de 0 de taille Nro et Ntheta  for i=1:30;  [maxval,idx]=max(Hough(:)); %recherche de la valeurs max du tableau de Hough et stockage de la valeur et de son indice  [row,col]=ind2sub(size(Hough),idx);%on stocke le numéro de l'indice de la valeur max en notant ses coordonnées en ligne/colonne  Hough\_s(row,col) = 1;%on sauvegarde cet indice dans un nouveau tableau  Hough(row,col) = 0;%on supprime cette valeur maximum pour pouvoir trouver la valeur max suivante  end    [l,k] = find(Hough\_s); %retourne un tableau avec les colonnes et lignes pour toutes valeurs non nulles dans le tableau Hough\_s |

On peut constater dans notre boucle que nous mettons à zéro la valeur dans l’espace de Hough correspondant à (l,k) afin de pouvoir supprimer cette valeur et de chercher la valeur maximale suivante. Les indices recherchés sont remplis dans un tableau différent, noté Hough\_s.

On créé ensuite une fonction [P1,P2]=DeuxPoints(ro,theta,H,W) renvoyant en sortie deux points extrêmes de l’image dont les coordonnées appartiennent à la droite définie par (ro, theta) afin de tracer cette dernière. On a alors en paramètres d’entrées les ro et theta correspondant aux valeurs maximales de l’espace de Hough. On place cette fonction dans une boucle affichant seulement les 30 droites correspondant aux 30 premières valeurs maximales.

Le code général correspondant est le suivant :

for p=1:30

%Renvoie deux points extrêmes de l'image appartenant a la droite définie par ro et theta

[P1,P2] = DeuxPoints(ro(p),theta(p),H,W);

figure(4),hold on;

plot([P1(1) P2(1)], [P1(2) P2(2)],'r');

title('Tracé des droites définies par ro et theta')

end

Le code de la fonction [P1,P2]=DeuxPoints(ro,theta,H,W) correspondant est le suivant :

function [P1,P2]=DeuxPoints(ro,theta,H,W)

a = cos(theta);

b = sin(theta);

x0 = ro\*a;

y0 = ro\*b;

x1 = (x0 + W\*(-b));

y1 = (y0 + H\*(a));

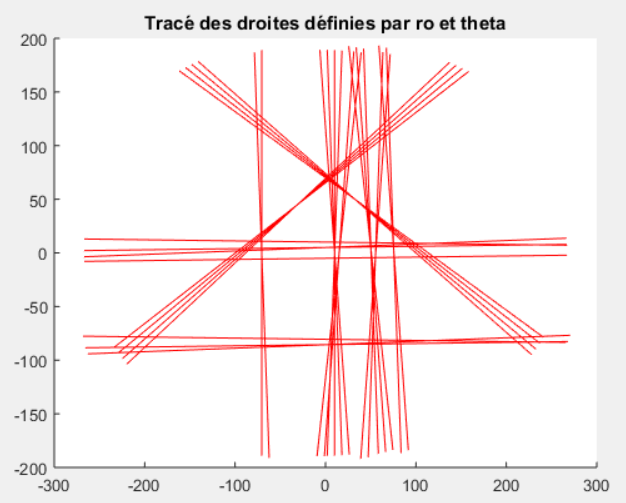
x2 = (x0 - W\*(-b));

y2 = (y0 - H\*(a));

P1 = [x1,y1];

P2 = [x2,y2];

end



On constate alors que ces droites décrivent approximativement la maison de l’image de base. Cela est normal car nous nous sommes basés sur les valeurs maximales de l’espace de Hough, et donc on peut quasiment reformer les lignes blanches de la maison. On n’a qu’une esquisse car il s’agit premièrement de droites, et non de segments, dues aux points extrêmes de l’image sélectionnés précédemment. Deuxièmement, le manque de précision sur le positionnement des droites vient du manque d’informations dans notre espace de Hough, deuxième point que l’on va essayer de rétablir avec une normalisation sur la transformation de Hough.

**Normalisation**

On décide dans cette partie de normaliser la transformation de Hough sur l’image de base. On va alors diviser cette dernière par une transformée de Hough sur une image de taille égale à celle de la maison mais cette fois composée exclusivement de 1. On programme alors de la même manière que la transformée écrite précédemment en changeant notre image de base.

Le code correspondant est le suivant :

%Normalisation

if normalisation==true

I\_norm=ones([H,W]); %Matrice de 1 de la taille de l'image

ind=find(I\_norm); %Récupère les indices des pixels non nuls

[i,j] = ind2sub(size(I\_norm),ind); %Conversion en indice ligne/colonne

Hough\_norm=zeros(Nro,Ntheta);

for n=1:length(ind)

%Relation liant x à j et y à i permettant la conversion de (i,j) en (x,y)

x= j(n) - x0;

y= -(i(n)-y0);

%Fonction retournant les sinusoïdes correspondant aux points (x,y)

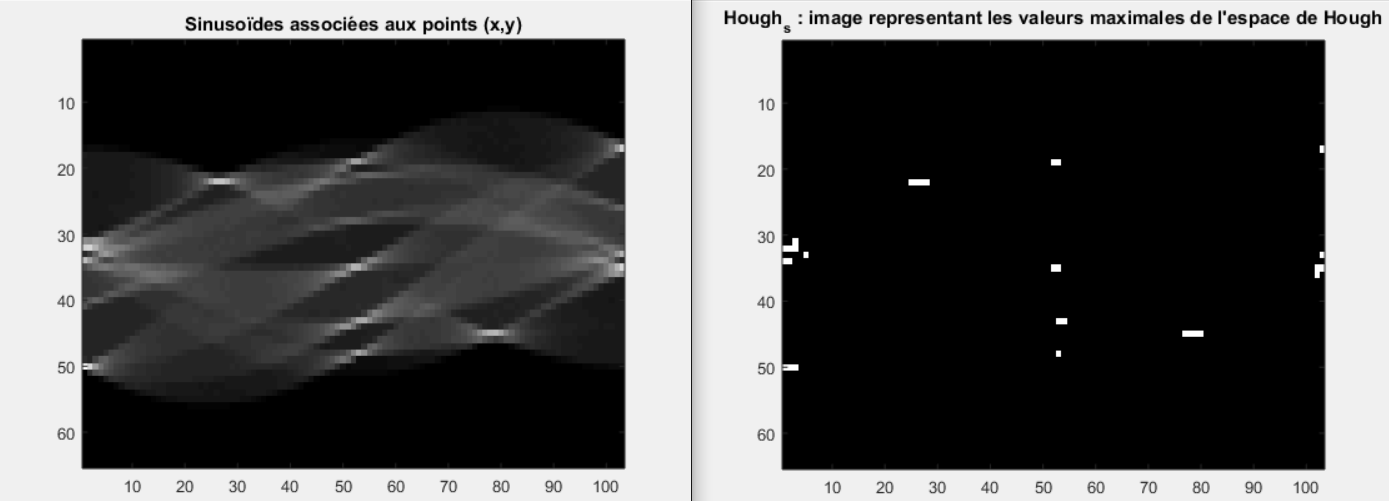
Hough\_norm=SinusHoughSpace2(Hough\_norm,x,y,ro\_max);

end

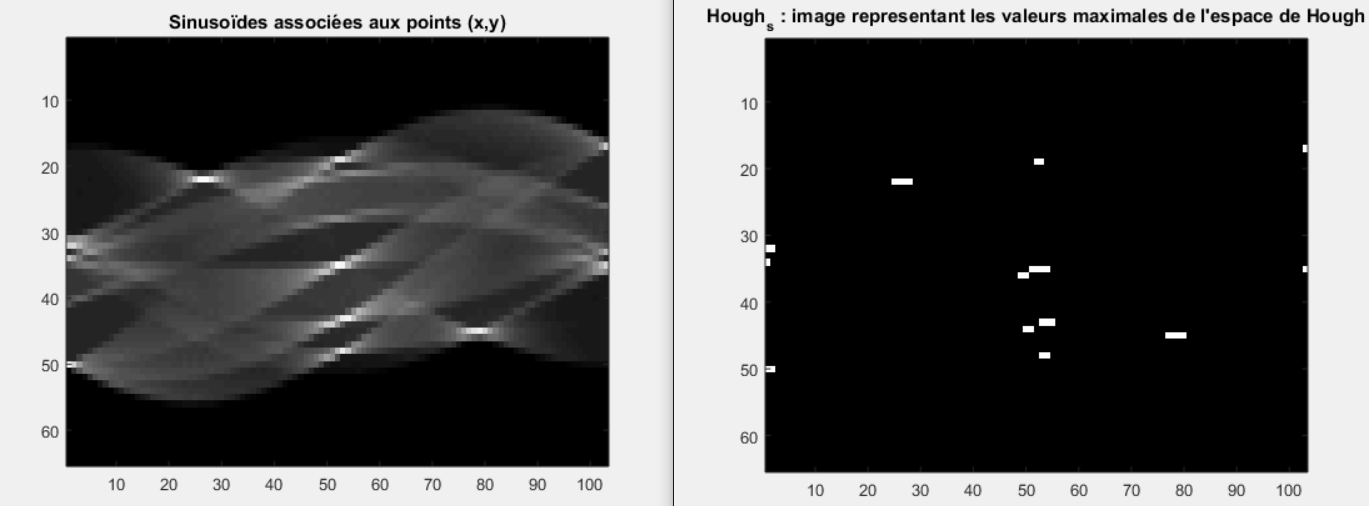
Hough=Hough./Hough\_norm;

end

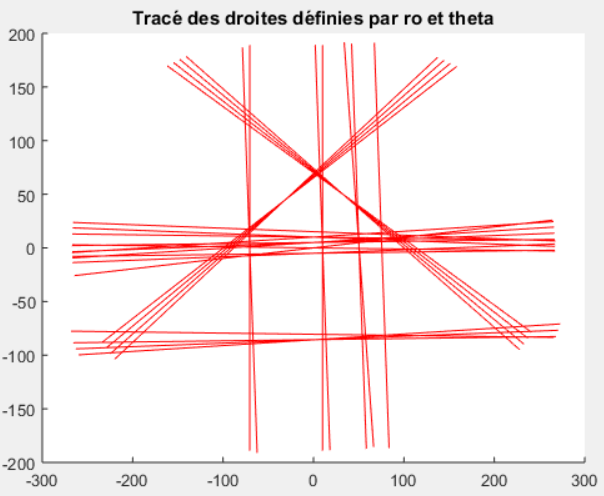
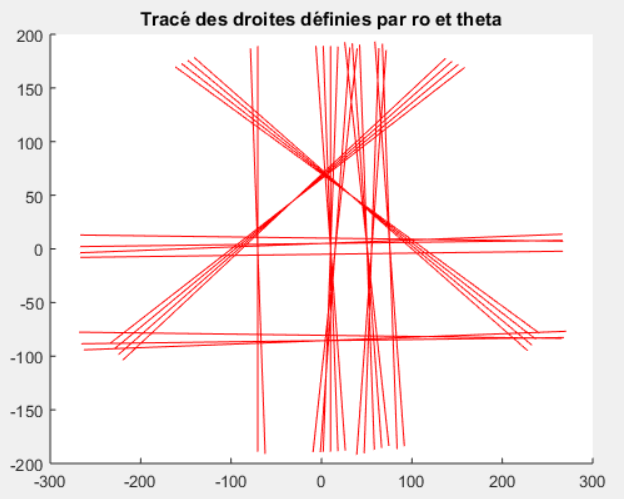
Après avoir effectuée la normalisation de cette transformation, nous pouvons observer une intensité au centre de l’espace de Hough plus importante notamment



Avant Normalisation



Après Normalisation



Avant Normalisation Après Normalisation

On peut alors constater sur l’image montrant les droitescorrespondant aux premières valeurs maximales de l’espace de Hough, que celles qui correspondent aux contours extérieurs de la maison (murs et sol) seront moins éparpillées et plus précises. Cette normalisation consiste à faire ressortir les valeurs maximales afin de gagner en précision.

**Conclusion**

On a vu dans ce TP comment implémenter la transformation de Hough afin de détecter des lignes présentes sur l’image. Pour cela, nous avons tracé différentes choses nous permettant d’accéder à la « reconstruction » de l’image par des droites décrites par des points de l’espace de Hough : transformée de Hough, sinusoïdes associées aux points, valeurs maximales de l’espace de Hough ainsi que les droites en découlant. Nous avons également effectué une normalisation nous permettant de gagner en précision sur certaines parties de cette reconstruction. Ce TP nous a permis de découvrir une technique permettant, à partir d’une image binaire, de détecter les lignes (et autres formes géométriques possibles mais non traitées dans ce TP) d’intensités maximales (valeur à 1) de l’image et de pouvoir les isoler.