

# Application de la propriété de pseudo-linéarité de la Transformée de Hough à l'amélioration de sa vitesse de calcul

**E.Duqueno**y, A.Taleb-Ahmed  
195, Rue du Pasteur MLK  
LASL - Université du Littoral - Calais, France  
Eric.Duquenoy@lasl-gw.univ-littoral.fr

J.P. Dubus  
I3D - Université de Lille 1 - Villeneuve d'Ascq, France

## 1 Introduction

La diminution du temps d'exécution d'une transformée de Hough passe par la réduction du nombre de points à traiter [6]. Dans ce but, [7][9] utilisent des informations *a priori* telles que la **direction** et l'**amplitude** du vecteur **gradient**, alors que [8][3] préconisent un **choix aléatoire** des points dans l'image binaire associé à une détection **simultanée** des maximums de l'espace transformé. Toutefois, ces types de méthodes se heurtent au problème du bruit dans l'image qui **accroît le nombre de points à traiter** augmentant ainsi le temps de calcul et le nombre de fausses détections.

Nous présentons donc, dans ce qui suit, une méthode générale d'accélération de la convergence de la transformée de Hough se basant sur une amélioration de la vitesse de scrutation de l'image. Pour cela, nous montrerons dans la section 2 que la transformée de Hough peut être *régie par une algèbre*, certes rudimentaire, mais suffisante pour justifier le choix de structures d'implantations logicielles (ou même matérielles) puis nous proposerons, section 3, un procédé s'apparentant à un *sous-échantillonnage spatial* de l'image binaire associé, section 4 à une *recherche adaptative du maximum dans l'espace transformé conduisant à une détection anticipée de ce maximum*. Nous décrivons enfin, section 5, des résultats démontrant les gains de performances obtenus.

## 2 Pseudo-linéarité de la T.H.

### 2.1 Transformée "one-to-many" ou "1-vers-m"

Nous cherchons à démontrer que la transformée de Hough est une *relation linéaire*  $\mathcal{HT}$  de l'ensemble  $I$  des points allumés de l'image binaire à traiter (espace de points de données), vers un ensemble  $P$  représentant l'espace des paramètres. Un élément  $M$  de l'ensemble  $I$  est caractérisé par son vecteur de coordonnées  $\underline{x}$  de dimension  $n$  à valeurs dans  $\mathbb{R}^n$  alors qu'un élément de l'ensemble  $P$  est caractérisé par un vecteur de paramètres  $\underline{p}$  de dimension  $m$  à valeurs dans  $\mathbb{R}^m$  et est associé à un accumulateur dans un espace d'accumulation  $A$ .

La relation  $\mathcal{HT}$  fait correspondre à chaque élément  $M_i$  de l'ensemble  $I$ , où  $i \in \{1 \dots \text{card } I\}$ , une courbe  $\mathcal{C}_i$ , c'est-à-dire un ensemble d'éléments dans l'ensemble d'arrivée  $P$ , et qui réalise l'incrémentatation des accumulateurs traversés. La transformée de deux éléments  $M_i$  et  $M_j$  de  $I$ , où  $i \in \{1 \dots \text{card } I\}$ , avec  $M_i \neq M_j$ , est définie comme la somme des deux courbes  $\mathcal{C}_i$  et  $\mathcal{C}_j$  dans l'ensemble d'arrivée  $P$ . Comme  $M_i$

et  $M_j \in I$ , si nous appelons  $I_i = \{M_i\}$  et  $I_j = \{M_j\}$ , nous pouvons écrire que :

$$\begin{cases} \mathcal{HT}(M_i) \equiv \mathcal{HT}(I_i) = \mathcal{C}_i \\ \mathcal{HT}(M_j) \equiv \mathcal{HT}(I_j) = \mathcal{C}_j \\ \mathcal{HT}(\{M_i, M_j\}) \equiv \mathcal{HT}(I_i \cup I_j) = \mathcal{C}_i + \mathcal{C}_j \\ \implies \mathcal{HT}(I_i \cup I_j) = \mathcal{HT}(I_i) + \mathcal{HT}(I_j) \end{cases} \quad (1)$$

Le résultat 1 permet de comprendre que la transformée d'un sous-ensemble de  $I$ , c'est-à-dire ici  $I_i \cup I_j$  peut se décomposer en la somme des transformées des singletons composant ce sous-ensemble. Par extension de cette propriété et en considérant une partition  $\mathfrak{P}(I) = \{I_1 \dots I_q\}$  telle que  $I = I_1 \cup I_2 \cup \dots \cup I_q$ ,  $\forall i$  et  $\forall j \in \{1 \dots q\}$  avec  $q \leq \text{card } I$ ,  $i \neq j$  et  $I_i \cap I_j = \emptyset$ , nous en déduisons :

$$\begin{aligned} \mathcal{HT}(I) &= \mathcal{HT}(I_1 \cup I_2 \dots \cup I_q) \\ \implies \mathcal{HT}(I_1 \cup I_2 \dots \cup I_q) &= \sum_{k=1}^q \mathcal{HT}(I_k) \end{aligned} \quad (2)$$

Le résultat de l'équation 2 s'énonce ainsi : *la transformée de Hough d'un ensemble est égale à la somme des transformées de chacun de ses sous-ensembles disjoints deux-à-deux et complémentaires* [1].

## 2.2 Transformée "many-to-one" ou "m-vers-1"

Dans le processus de transformation de Hough utilisant une méthode de type "many-to-one", la valeur du vecteur de paramètres  $\mathbf{p}$  dans l'espace  $P$  est calculée à partir d'un élément  $M$  de l'ensemble  $I \times I \dots \times I$ .  $M$  est, de ce fait, un sous-ensemble d'éléments distincts de  $I$ , chacun étant caractérisé par son vecteur de coordonnées  $\mathbf{x}$ . L'espace de départ est donc constitué d'éléments pris dans  $I \times I \dots \times I$  alors que l'espace d'arrivée  $P$  reste inchangé. De plus, il ne s'agit plus ici de calculer une courbe  $\mathcal{C}_i$ , ensemble d'éléments de l'espace d'arrivée  $P$ , mais un élément unique de cet espace.

La relation  $\mathcal{HT}$  fait correspondre à chaque élément  $M = (M_1 \dots M_\gamma)$  de l'ensemble  $I \times I \dots \times I = I^\gamma$ ,  $\gamma$  représentant la dimension de l'espace transformé c'est-à-dire le nombre de paramètres recherchés, un point ayant pour vecteur de paramètres  $\mathbf{p}$  dans l'espace d'arrivée  $P$ . Ainsi, la relation 2 reste toujours valable, à condition de considérer  $I^\gamma$  comme ensemble de départ.

## 2.3 Conséquences

Nous pouvons déduire, à partir de la propriété de "pseudo-linéarité" démontrée par l'équation 2, deux conséquences sur la méthode à utiliser pour calculer de manière optimale la transformée de Hough :

1. **La possibilité de décomposer le calcul de la transformée complète** d'une image binaire en **une somme de transformées** de sous-ensembles de points.
2. **Chaque transformée de la décomposition peut être calculée indépendamment**, ce que nous exploiterons dans la méthode proposée dans la section suivante.

# 3 Accélération par sous-échantillonnage spatial

## 3.1 Formalisme

Le principe de sous-échantillonnage utilisé dans notre méthode, consiste en une décomposition de l'image binaire initiale  $I$  de dimensions  $N * N$  en un ensemble de sous-images  $I_{a,b}^{\mathcal{O}_e}$  où le paramètre  $\mathcal{O}_e$  est l'**ordre de sous-échantillonnage** et le couple  $(a, b)$  la référence de la sous-image, et tel que  $I = \cup I_{a,b}^{\mathcal{O}_e}$  et  $\cap I_{a,b}^{\mathcal{O}_e} = \emptyset$ , c'est-à-dire respectant les conditions de l'équation 2. Les points des sous-images sont choisis selon la règle suivante :

$$M(x, y) \in I_{a,b}^{\mathcal{O}_e} \Leftrightarrow M(x, y) \in I(N, N) \text{ avec } x = \mathcal{O}_e \cdot k + a \text{ et } y = \mathcal{O}_e \cdot k + b \quad (3)$$

$$\text{avec : } \mathcal{O}_e \in \mathbb{N}, a \text{ et } b \in [0, \mathcal{O}_e[, k \in \left[0, \frac{N}{\mathcal{O}_e} - 1\right] \text{ pour } \mathcal{O}_e \neq 0$$

**Exemple 1** : Pour  $\mathcal{O}_e = 2$  et  $N = 256$ , nous obtenons que  $k \in [0, 127]$ ,  $a$  et  $b \in [0, 2[$ ,  $x = 2 * k + a$  et  $y = 2 * k + b$

L'intérêt d'une telle décomposition est d'accroître la **vitesse de scrutation de l'image**, ce qui conduit la transformée à prendre plus rapidement en compte l'ensemble du ou des objets contenus dans l'image. **Nous augmentons ainsi la perception globale de l'image par la méthode de détection de formes.**

La transformée de  $I$  étant, selon le résultat 2, égale à la somme des transformées de ses sous-images  $I_{a,b}^{\mathcal{O}_e}$ , le sous-échantillonnage spatial n'influence pas le résultat final, c'est-à-dire l'espace des paramètres. Comme les images traitées sont binaires, nous pouvons écrire que :

$$I = \cup I_{a,b}^{\mathcal{O}_e} \implies \mathcal{HT}(I) = \sum \mathcal{HT}(I_{a,b}^{\mathcal{O}_e}) \quad (4)$$

de même, pour la transformée "many-to-one" pour une recherche de formes à deux paramètres :

$$\begin{aligned} I^2 &= I \times I \\ &= \cup I_{a,b}^{\mathcal{O}_e} \times \cup I_{a,b}^{\mathcal{O}_e} \\ &= \bigcup_{\substack{i=0 \dots \mathcal{O}_e-1 \\ j=0 \dots \mathcal{O}_e-1}} (I_{i,j}^{\mathcal{O}_e})^2 \cup \left\{ (I_{0,0}^{\mathcal{O}_e} \times I_{0,1}^{\mathcal{O}_e}) \cup \dots \cup (I_{\mathcal{O}_e-1, \mathcal{O}_e-1}^{\mathcal{O}_e} \times I_{\mathcal{O}_e-1, \mathcal{O}_e-2}^{\mathcal{O}_e}) \right\} \\ &\implies \mathcal{HT}(I \times I) = \sum \mathcal{HT}((I_{i,j}^{\mathcal{O}_e})^2) + \mathcal{HT}(I_{0,0}^{\mathcal{O}_e} \times I_{0,1}^{\mathcal{O}_e}) + \dots \mathcal{HT}(I_{\mathcal{O}_e-1, \mathcal{O}_e-1}^{\mathcal{O}_e} \times I_{\mathcal{O}_e-1, \mathcal{O}_e-2}^{\mathcal{O}_e}) \quad (5) \end{aligned}$$

En associant cette méthode à un **détecteur de maximum** comme celui proposé dans la partie suivante, il est également possible de stopper le calcul de la transformée avant son terme et donc, d'**accélérer l'obtention du pic**.

## 4 Détection de maximum par recherche adaptative

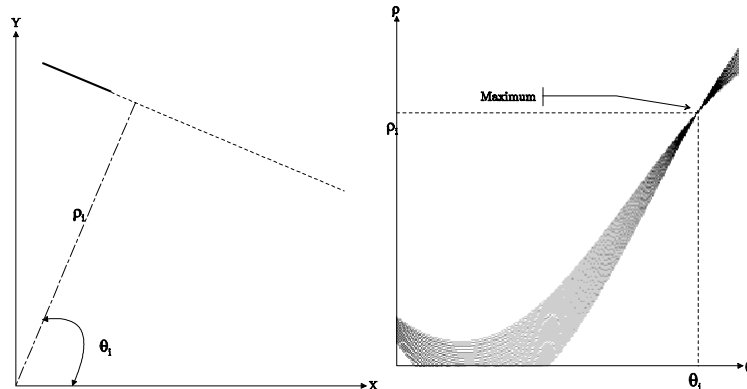
### 4.1 Introduction

Le pic de l'espace d'accumulation est caractérisé par sa **position**, qui détermine la valeurs des paramètres recherchés, et sa **valeur**, c'est-à-dire le nombre de votes accumulés. Dans une transformée classique, les phases d'accumulation et de détection de pics sont exécutées **successivement**. Cette manière de procéder engendre alors un certain nombre de défauts que nous allons tenter de cerner.

### 4.2 Défauts des approches classiques

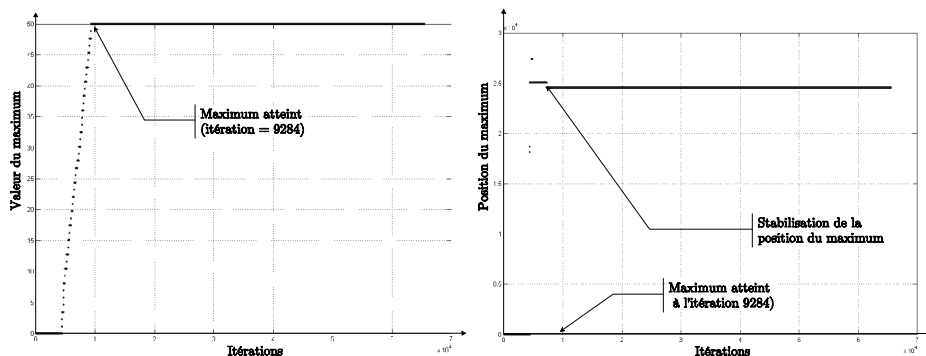
Le segment de droite de la figure 1a est détecté par une transformée standard [4]. Si l'on suppose que l'extraction des points s'effectue de la gauche vers la droite et de haut en bas, lorsque le dernier point du segment de droite aura été traité, le pic de l'espace transformé, figure 1b, **aura atteint son maximum et n'évoluera donc plus pendant le reste de la scrutation soit plus du 3/4 de la durée totale d'exécution** du traitement de l'image. Dans la méthode classique, vote et recherche de maximum s'exécutant séquentiellement et non **simultanément**, ce défaut est rédhibitoire.

La figure 1c représente l'évolution de la valeur du maximum lors du traitement de l'image de la figure 1a. C'est à l'itération 9284, pour un nombre total égal à 65536, que le maximum atteint sa valeur définitive de 50. En termes relatifs, cela conduit à considérer que **si le calcul de la transformée avait été stoppé après l'itération 9284, cela aurait apporté un gain en vitesse de traitement de l'ordre de 80%**.



(a) Segment de droite placé dans le premier quart haut de l'image

(b) Espace transformé correspondant à la figure a



(c) Evolution de la valeur du maximum en fonction du nombre d'itérations

(d) Evolution de la position du maximum en fonction du nombre d'itérations (pour la figure 1a)

Figure 1: Le segment de droite (a) est composé de 50 points. L'amplitude du pic de la figure (b) est égale à 50.

Cet exemple montre donc que **la mesure du maximum conjointement au calcul de l'espace transformé peut permettre d'envisager un gain de temps appréciable.**

Une autre observation peut être effectuée, cette fois sur l'évolution de la **position** du maximum. En effet, dans le cas de la figure 1d, nous constatons que la position du maximum (en termes d'indices) dans l'espace transformé se stabilise **avant** sa valeur. Cette observation laisse donc à penser qu'il est possible d'anticiper la détection du maximum par le suivi de l'**évolution de sa position**. Dans ce cas, la détection du pic n'est plus liée à une valeur d'amplitude elle-même fortement dépendante du contexte de l'image (nombre d'objets, occultation, bruit), ce qui la rend nettement plus robuste qu'un simple seuillage. Dans le paragraphe suivant, nous allons montrer comment mettre en place une détection de pic anticipée qui soit adaptative, c'est-à-dire indépendante du contexte.

### 4.3 Recherche adaptative du maximum

#### 4.3.1 Limitation des méthodes existantes

Il a été proposé dans [5] une recherche de maximum simultanément à la phase de vote. Le principe de cette procédure consiste, après chaque vote, à comparer le maximum courant à un seuil préétabli, de valeur en général très faible. Lorsque ce seuil est atteint, une transformée inverse est immédiatement calculée

permettant d'extraire les points ayant participé au vote. Cette méthode est cependant mise en échec dans de nombreux cas :

- La présence de bruit dans l'image, en diminuant le rapport signal/bruit, conduit à l'obtention d'un maximum dans l'espace transformé qui peut fluctuer en **valeur** et en **position**.
- Le même segment de droite placé dans des contextes différents de fond ou d'environnement ne conduira pas forcément à la même valeur de pic.
- Le choix d'un seuil fixe nécessite une connaissance *a priori* du contenu ou de la nature des images à traiter.
- Un seuil trop faible entraîne des fausses détections alors qu'un seuil trop élevé empêche la détection de segments courts.

#### 4.3.2 Solution proposée

Pour qu'une détection de maximum anticipée soit réellement efficace, il est nécessaire d'utiliser un critère qui permette **l'arrêt des calculs dès lors que la valeur du maximum est obtenue avec certitude**. Cependant, la durée de convergence vers ce maximum dépend d'un certain nombre de paramètres :

- le nombre de points de l'image à traiter. Plus ce nombre est élevé, plus la durée de convergence augmente. L'utilisation conjointe de la méthode de sous-échantillonnage présentée section 3 permet de réduire cette durée : plus l'ordre de sous-échantillonnage est élevé, plus faible est la durée de convergence. Cependant, le corollaire est alors une augmentation de l'imprécision.
- le délai permettant d'affirmer que la convergence est atteinte. Ce paramètre dépend du niveau de bruit dans l'image. Le délai est exprimé en nombre d'itérations.

La solution que nous proposons se décompose en deux phases :

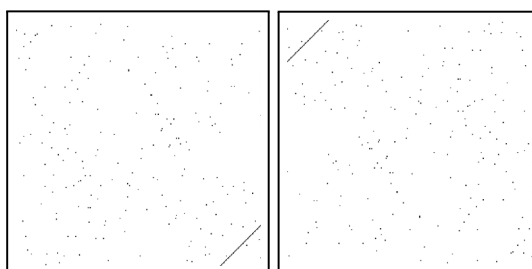
1. **Seuillage préliminaire du maximum** : lorsque la valeur de maximum atteint un seuil prédéfini, nous passons à la seconde phase. Cette première phase permet d'ajuster la méthode à la nature de l'image : elle n'est donc pas obligatoire, le seuil pouvant être choisi aussi bas que possible. Il ne s'agit en fait que d'initialiser la seconde phase.
2. **"Verrouillage" du maximum** : la durée (nombre d'itérations) de stabilité de la position est mesurée. Si cette durée est supérieure à un seuil prédéfini, alors nous pouvons estimer qu'il s'agit du pic recherché et le calcul de la transformé peut être stoppé.

#### 4.3.3 Conclusion

Notre méthode est donc **adaptative** puisque d'une part le premier seuillage peut être remis en question par la seconde phase de la méthode et d'autre part nous utilisons l'information de position du maximum et non celle de valeur ce qui la rend indépendante de nombre de points qui composent les formes à détecter et du bruit présent dans l'image.

## 5 Résultats

Nous avons appliqué notre méthode d'accélération à deux types de transformées de Hough concernant pour la première, la détection de droites selon la méthode standard de *Duda et Hart* [4] et, pour la seconde, la localisation du centre d'une ellipse selon une méthode proposée par *Yuen et al.* dans [10]. Nous allons, dans ce qui suit, présenter les résultats obtenus.



(a) *Segment de droite*  
 $(\theta, \rho) = (45, 150)$       (b) *Segment de droite*  
 $(\theta, \rho) = (45, -150)$

	Temps	$\rho$	$\theta$	Pic
Standard	1.96s	151.65	44.78°	44
Avec maximum anticipé	1.48s	151	45.13°	44

(c) Résultats correspondants à la figure (a)

	Temps	$\rho$	$\theta$	Pic
Standard	1.96s	-148.1	44.96°	44
Avec maximum anticipé	0.62s	-148.1	44.96°	33

(d) Résultats correspondants à la figure (b)

Figure 2: *Mise en évidence de l'influence de la détection de maximum anticipée sur les temps d'exécution. Les deux images ont traitées avec la méthode standard de Duda et Hart [4].*

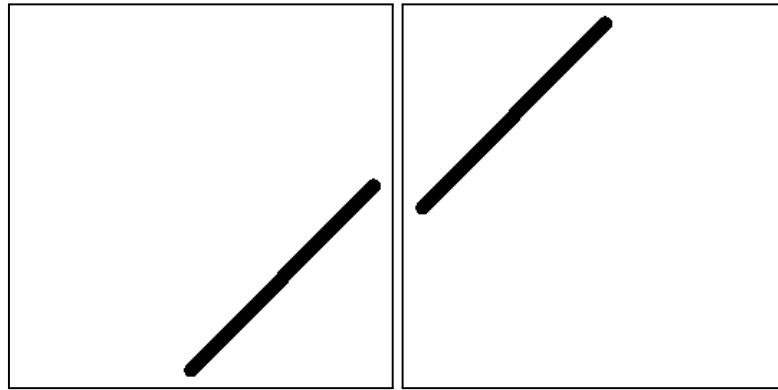
## 5.1 Application à la détection de droite

### 5.1.1 Mise en évidence de la détection anticipée de maximum

Pour la détection de droites, deux images de test ont été utilisées. Chacune d'elles comporte un segment d'une taille de 44 pixels (épaisseur d'un pixel) et d'orientation  $\theta$  égale  $= 45^\circ$ , seule la valeur de  $\rho$  (150 et  $-150$ ) étant différente et choisie de manière à obtenir un segment placé tantôt en haut à gauche, tantôt en bas à droite dans l'image (figures 2a et 2b). De plus, 200 points ont été ajoutés de manière aléatoire à l'image. Les résultats obtenus démontrent l'efficacité de notre principe de détection anticipée : si dans le cas de la figure 2a, dont les résultats sont présentés sur le tableau 2c, le gain en temps de traitement n'est que de l'ordre de 25%, pour la seconde image (figure 2b et tableau 2d), il s'agit d'une **accélération de presque 70%**. L'explication de cet accroissement de la rapidité de traitement de deux images au contenu semblable s'explique par la disposition géographique du segment à détecter : le balayage de l'image s'effectuant de haut en bas et de gauche à droite lors du traitement, le segment de l'image 2b est détecté plus rapidement que celui de la figure 2a. Les valeurs de  $\rho$  et  $\theta$  sont obtenues avec une précision bien inférieure à 1% par rapport à la méthode standard.

### 5.1.2 Mise en œuvre du sous-échantillonnage spatial.

De manière à mettre en évidence l'efficacité du sous-échantillonnage spatial, nous utilisons ici encore deux images de test ne comportant qu'un seul segment de droite à  $45^\circ$  (figures 3a et 3b). Cependant, à la différence des images précédentes, les segments ont une largeur de plusieurs pixels. Dans cette situation, la transformée de Hough détecte une droite qui n'est pas systématiquement dans la direction principale (figures 3e et 3f). Nous présentons dans les tableaux 3c et 3d, les résultats obtenus d'une transformée



(a) Segment de droite épais  
 $(\theta, \rho) = (45, -80)$

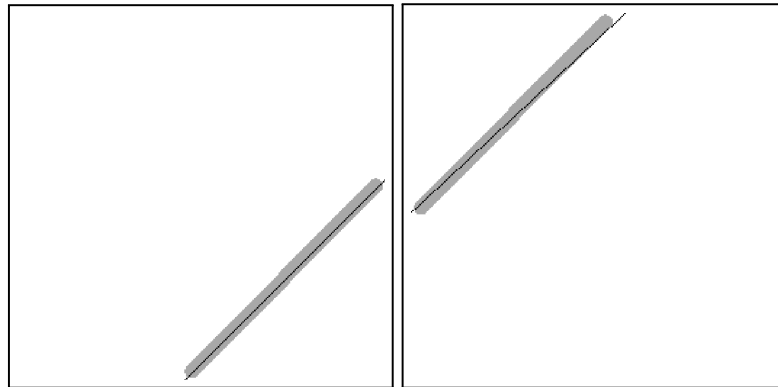
(b) Segment de droite épais  
 $(\theta, \rho) = (45, 80)$

	Temps	$\rho$	$\theta$	Pic
Standard	8s	81.6	44.96°	135

(c) Résultats d'une transformée standard de la figure (a)

	Temps	$\rho$	$\theta$	Pic
Standard	8s	-78.4	43°	73

(d) Résultats d'une transformée standard de la figure (b) (Erreur de 2° sur  $\theta$ )



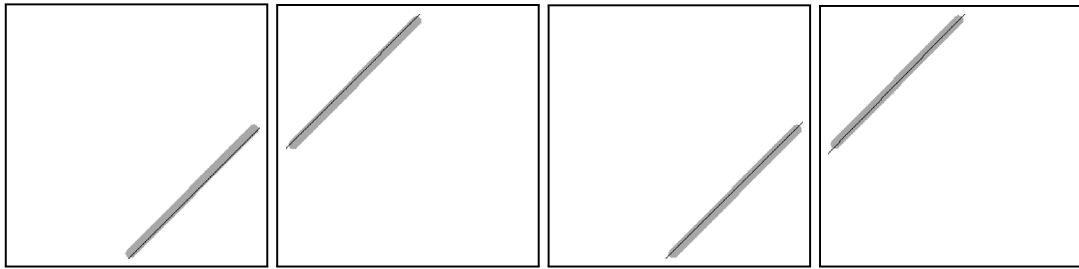
(e) Segment détecté

(f) Segment détecté

Figure 3: Transformation standard pour des segments épais (Duda et Hart [4])

standard. Les temps de calculs sont bien évidemment identiques puisque la transformée standard effectue la détection de pic après une scrutation totale de l'image. Pour la figure 3f, une imprécision de 2° sur l'angle  $\theta$  est à noter.

Nous appliquons ensuite notre principe de sous-échantillonnage spatial associé à la détection de maximum anticipée. Les résultats sont présentés sur les figures 4a à 4d. Les performances obtenues, tableaux 4e et 4f, démontrent l'efficacité de notre principe de sous-échantillonnage : le gain en rapidité est en moyenne de 75%, et ceci sans que la précision ne soit diminuée.



(a) Traitement du segment de la figure 3a pour  $\mathcal{O}_e = 2$

(b) Traitement du segment de la figure 3b pour  $\mathcal{O}_e = 2$

(c) Traitement du segment de la figure 3a pour  $\mathcal{O}_e = 4$

(d) Traitement du segment de la figure 3b pour  $\mathcal{O}_e = 4$

(a) et (c)	Temps	$\rho$	$\theta$	Pic
$\mathcal{O}_e = 2$	2.6s	-83	44.9°	68
$\mathcal{O}_e = 4$	0.57s	78.7	44.96°	14
Standard	8s	81.6	44.96°	135

(b) et (d)	Temps	$\rho$	$\theta$	Pic
$\mathcal{O}_e = 2$	1.4s	-82.3	45.1°	34
$\mathcal{O}_e = 4$	2.2s	-79.1	45.6°	19
Standard	8s	-78.4	43°	73

(e) Comparaison des résultats correspondants aux figures (a) et (c)

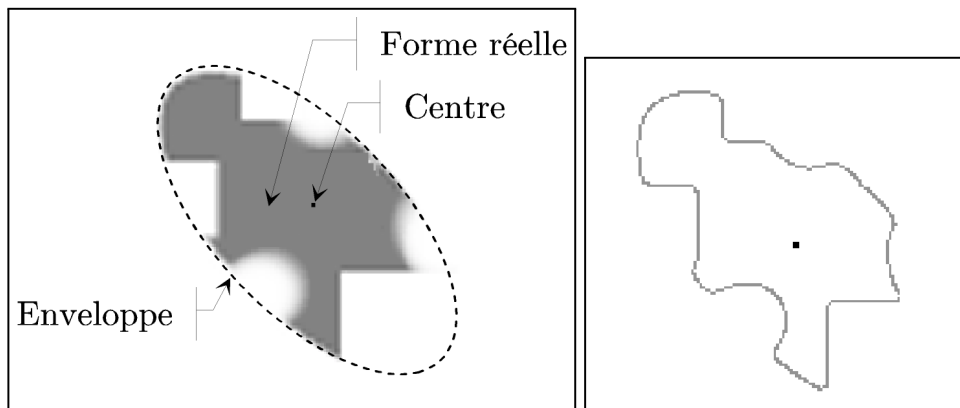
(f) Comparaison des résultats correspondants aux figures (b) et (d)

Figure 4: Mise en œuvre du sous-échantillonnage spatial pour la détection segments épais.

## 5.2 Application à la localisation de centres d'ellipses.

### 5.2.1 Introduction

Dans de nombreux cas où les formes ellipsoïdales ne présentent pas de symétrie par rapport au centre, la méthode de *Tsuji et al.* [7] est inopérante car elle se base sur la sélection de points dont les tangentes sont parallèles. Il est alors nécessaire de mettre en œuvre une méthode basée sur d'autres propriétés de l'ellipse. La méthode proposée par *Yuen et al.* dans [10] s'appuyant sur la notion de pôles et de polaires, présente un coût calculatoire plus élevé que la précédente dans le sens où, pour chaque couple de points de l'espace de départ, il faut incrémenter les accumulateurs de l'espace transformé traversés par une droite passant par le pôle et le milieu de la polaire. L'image de test utilisée est présentée figure 5.



(a) Ellipse dont les contours ne présentent aucune symétrie par rapport à son centre

(b) Contour de l'ellipse (a)

Figure 5: Image de test utilisée (256 \* 256 pixels).



### 5.3 Mesure de référence : méthode de Yuen sans accélération

Dans un premier temps, la méthode de calcul de centre d'ellipse par transformée de Hough est mise en œuvre de manière classique, c'est-à-dire sans que soient utilisés nos principes d'accélération développés dans les paragraphes précédents. Le résultat obtenu est ensuite employé comme base de comparaison. Dans un deuxième temps, nous appliquons nos deux principes d'accélération. Le temps de calcul obtenu pour la méthode de *Yuen et al.* est de  $42s$  et ceci pour un nombre de points traités de 390. Rappelons que le temps de calcul d'une transformée de Hough est proportionnel au nombre de points de l'image binaire à traiter. Dans ce cas précis de [10], ce temps de calcul est même proportionnel à  $n(n-2)/2$  où  $n$  représente le nombre de points. La valeur élevée de temps de calcul est liée à la complexité de la méthode de *Yuen* qui nécessite le calcul de la droite passant par le pôle et la polaire puis l'incrémentation des accumulateurs traversés par cette droite.

La méthode de *Yuen* utilise l'information de direction de gradient pour calculer les directions des tangentes au contour de manière à établir la position du pôle de l'ellipse (intersection des tangentes). L'utilisation d'une information issue du calcul d'un gradient directionnel présente néanmoins le risque d'introduire des erreurs dans le processus de transformation étant donné que la direction de gradient intervient dans le calcul des tangentes. En effet, une erreur sur la direction du gradient entraîne une erreur sur le calcul du pôle et donc sur la droite passant par le pôle et la polaire c'est-à-dire finalement sur la localisation du centre. Cela explique la légère imprécision constatée dans certains cas dans les mesures qui suivent.

Nous allons maintenant mettre en évidence l'influence du principe de sous-échantillonnage sur la vitesse de balayage de l'image. La conséquence de cet accroissement de la célérité de scrutation apparaît tout particulièrement dans le cas où le seuil de détection du pic dans l'espace transformé ou bien le seuil de délai de stabilité de ce pic est choisi relativement bas. En effet, dans cette situation, il est possible d'obtenir de fausses détections dues à l'absence d'analyse globale de l'image. Le sous-échantillonnage va permettre d'étendre l'analyse à la totalité de l'image c'est-à-dire d'accroître la perception globale de l'image par la transformée de Hough. C'est ce que nous constatons dans le cas de l'application de la méthode de *Yuen* à la forme ellipsoïdale dont les résultats, présentés figure 6, sont résumés dans le tableau 1 où les valeurs en gras sont inférieures à celles de référence ( $45s$ ) et les valeurs en italiques correspondent à un centre détecté erroné. Nous constatons en effet qu'à partir d'un ordre de sous-échantillonnage  $\mathcal{O}_e = 2$ , la méthode fournit un résultat proche du centre recherché pour un gain de performances en rapidité supérieur à 80%.

Forme ellipsoïdale Seuil = 500	Yuen	Erreur en pixels
$\mathcal{O}_e = 0$	<i>0.52s</i>	<i>26</i>
$\mathcal{O}_e = 1$	<i>1.7s</i>	<i>65</i>
$\mathcal{O}_e = 2$	<b>6s</b>	<b>1</b>
$\mathcal{O}_e = 3$	<b>4s</b>	<b>3</b>

Table 1: Temps de calcul de la transformée de Yuen avec application à la forme ellipsoïdale de la méthode de **sous-échantillonnage spatial** (seuil maximum de stabilité de 500).

Le gain de performance en vitesse se fait nettement sentir, en particulier pour  $\mathcal{O}_e = 2$  et  $\mathcal{O}_e = 3$  par rapport à l'application de la méthode sans sous-échantillonnage.

## 6 Conclusion

Nous avons mis en évidence dans cette section la possibilité de présenter la transformée de Hough comme une opération linéaire. Cette propriété nous a permis de proposer une méthode de décomposition de l'image à traiter qui accroît la vitesse de scrutation et élargissant ainsi sa perception globale par la transformée, cette démarche conduisant également à éviter les fausses détections.

L'objectif étant d'accroître la rapidité de calcul de la transformée de Hough, nous avons choisi de réduire le nombre de points auxquels la transformée est appliquée, tout en augmentant la perception globale de l'image, par le principe de sous-échantillonnage combiné à un critère d'arrêt du calcul de la transformée, basé sur une observation de la stabilité en position du pic de l'espace transformé.

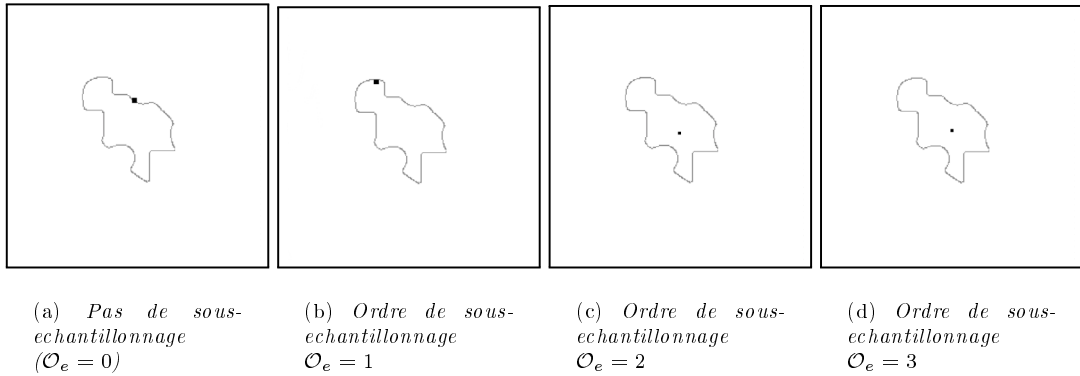


Figure 6: *Détection du centre de la forme ellipsoïdale de la figure 5 avec application des principes de sous-échantillonnage spatial et de détection de maximum anticipée.*

Nous avons montré les gains de performances que peuvent apporter nos deux principes d'accélération de la transformée de Hough que sont la **détection de maximum anticipée** et le **sous-échantillonnage spatial**. Il nous reste à mettre en place un critère qui permette le choix automatique de l'indice de sous-échantillonnage spatial.

## References

- [1] E. Duquenoy. *Accroissement de la Vitesse de Convergence de la Transformée de Hough et Contribution à la Détection de Contours par Fenêtre Ductile*. PhD thesis, Université du Littoral - Côte d'Opale, 1998.
- [2] S. Ben Yacoub. *Extraction hiérarchique robuste d'attributs géométriques dans une image numérique*. PhD thesis, Université Claude Bernard, Lyon I, 1995.
- [3] J.R. Bergen and H Shvaytser. A probabilistic algorithm for computing hough transforms. *Journal of algorithms*, 12(4):639–656, 1991.
- [4] R.O. Duda and P.E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, pages 11–15, 1972.
- [5] H. Kalviainen, E. Oja, and L. Xu. Motion detection using randomized hough transform. In *Proceedings of 7th Scandinavian Conference on Image Analysis*, pages 72–79, Aalborg, Denmark, August 1991.
- [6] N. Kiryati and Bruckstein A. On navigating between friends and foes. *IEEE Transactions on pattern analysis and machine intelligence*, 13(6):602–606, 1991.
- [7] S. Tsuji and F. Matsumoto. Detection of ellipses by a modified hough transformation. *IEEE Transactions On Computers*, c-27(8):777–781, 1978.
- [8] Lei Xu and Erkki Oja. Randomized hough transform (rht) : basic mechanisms, algorithms, and computational complexities. *CVGIP. Image understanding*, 57(2):131–154, 1993.
- [9] J.H. Yoo and I.H. Sethi. An ellipse detection method from the polar and pole definition of conics. *Pat. Recog.*, 26(2):307–315, 1993.
- [10] H.K. Yuen, J Illingworth, and J. Kittler. Detecting partially occluded ellipses using the hough transforme. *image and vision computing*, 7(1):31–37, 1989.