

# Méthode de détection de l'inclinaison des documents arabes imprimés basée sur la réduction des points et la transformée de Hough

A. Sehad\*, L. Mezai\*, L. Sekkai\*, M. Cheriet\*\*

*\*Division Architecture des Systèmes et Multimédia  
Centre de Développement des Technologies Avancées Cité 20 Août, BP 11, Baba Hassen, Alger*

Sehad@cdta.dz

l\_mezai@yahoo.fr

poi\_lamia@yahoo.fr

*\*\* Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle  
École de Technologie Supérieure, Canada*

mohamed.cheriet@gpa.etsmtl.ca

**Résumé:** Dans cet article, nous présentons une nouvelle méthode de détection de l'inclinaison de documents arabes imprimés. La transformée de Hough est utilisée et adaptée en tolérant un angle d'inclinaison de 0° à 180°. Nous avons considéré les connaissances à priori de l'écriture arabe imprimée qui est cursive par nature. En effet, en exploitant des tracés stables et homogènes, à savoir les liaisons qui relient les caractères imprimés, et en les modélisant par le centre de leur boîte englobante, notre approche permet une meilleure approximation de l'inclinaison du texte imprimé, tout en réduisant le temps de calcul. Des expérimentations sur cent documents imprimés et variés montrent des résultats prometteurs avec des temps d'exécutions tolérables dans le contexte d'applications réelles.

**Mots clés:** Détection de l'inclinaison, détection des liaisons, documents arabes imprimés, transformée de Hough.

## 1 Introduction

L'analyse et la reconnaissance de documents a pour but de convertir un document sous format papier vers un format électronique compréhensible et réutilisable. Le document papier, une fois converti sous forme électronique, permet une recherche par le contenu, un transfert très rapide, un archivage et une gestion beaucoup plus aisée. Par contre, le document sous forme papier permet à l'utilisateur de disposer d'une plus grande autonomie, d'une possibilité de lecture détaillée et plus confortable.

L'un des problèmes le plus fréquent dans un système d'analyse et de reconnaissance de documents est la détermination de l'inclinaison du document. En général, l'inclinaison des images est provoquée essentiellement soit par un mauvais positionnement des pages lors de la saisie optique, soit par une mise en page fantaisiste et irrégulière de l'auteur.

Parmi les techniques de détection de l'angle d'inclinaison les plus utilisées, nous pouvons citer la méthode de projection, la transformée de Hough et la méthode des k-plus proches voisins.

La méthode de projection (Bagdanov & al., 1997), (Belaid & al., 1992), (Bunke & al., 1997), (Dance & al., 1999), (Kavallieratou & al., 2002) est basée sur le calcul de l'histogramme horizontal de l'image du document pour chaque angle appartenant à l'intervalle de détection. Cette méthode est facile à implémenter, appropriée pour des documents à structure simple, mais elle n'est pas appropriée pour des documents complexes contenant des graphiques ou des photographies et l'angle d'inclinaison est compris entre -10° et +10°.

La transformée de Hough (Amin & al., 2001), (Berlger & al., 1998), (Hull & al., 1998), (Jain & al., 1996), (Le & al., 1994), (Perantonis & al., 1999), (Parker & al., 1996), (Yin, 2001), (Yu & al., 1999) est une technique de détection des lignes et des courbes dans une image. Elle est utilisée aussi pour détecter l'angle d'inclinaison avec un intervalle de détection

compris entre  $0^\circ$  et  $180^\circ$ . Cette méthode est exacte, robuste et appropriée pour des documents multi-colonnes, mais elle nécessite un espace mémoire important et un temps de traitement prohibitif.

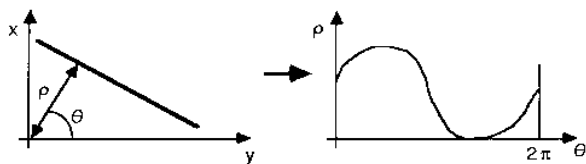
La méthode des k-plus proches voisins (Antonacopoulos & al., 1997), (Kwag & al., 2002), (O’Gorman, 1993), (Sababakhsh & al., 2000) consiste à calculer l’orientation de la ligne qui relie chaque deux composantes connexes voisines, les orientations sont représentées dans un histogramme. L’angle d’inclinaison du document est déterminé à partir du pic de l’histogramme. Cette méthode est appropriée pour des documents multi-colonnes, elle détecte plusieurs angle d’inclinaison, l’intervalle de détection est illimité par contre le temps de traitement est très important et elle est sensible au bruit et à l’écriture cursive particulièrement les documents arabes.

Dans cet article, nous proposons une nouvelle méthode de réduction de nombre de points à appliquer sur la transformée de Hough et cela afin de diminuer le temps de traitement et de garantir une exactitude dans la détection de l’angle d’inclinaison. Pour cela, nous avons utilisé une caractéristique de l’écriture arabe : les liaisons reliant les caractères arabes car elles ont presque la même hauteur et la même largeur et elles sont alignées sur la même ligne.

Dans la section suivante, nous présentons la transformée de Hough. Ensuite, dans la section 3, nous abordons les caractéristiques de l’écriture arabe. La section 4 présente les détails de notre approche. Les tests et les résultats sont traités dans la section 5. Enfin nous terminons par une conclusion.

## 2. Détection de l’inclinaison par la transformée de Hough

La transformée de Hough (Amin & al., 1996), (Bunke & al., 1997), (Hull & al., 1998), est une méthode qui permet de détecter les formes paramétriques dans des images, en particulier elle est utilisée pour la détection des lignes dans une image. Chaque ligne peut être définie par l’équation suivante :  $\rho = x \cdot \cos\theta + y \cdot \sin\theta$  tel que  $\rho$  est la distance de la normale entre l’origine et la ligne, et  $\theta$  est l’angle entre la normale et l’axe des X.

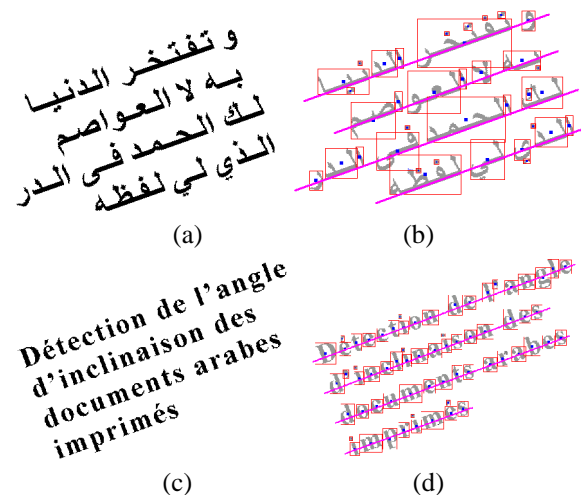


**Figure 1.** Représentation d’une droite dans l’espace de Hough.

Dans la pratique, la transformée de Hough est implémentée par un vecteur accumulateur qui est paramétré par les valeurs de  $\rho$  et  $\theta$ . Pour chaque pixel noir, la valeur de  $\rho$  est calculée en faisant varier  $\theta$  dans l’intervalle  $[0^\circ, \pi]$ , le contenu de la cellule

d’indice  $(\rho, \theta)$  est incrémentée. Après avoir traité toute l’image, le maximum  $(\rho_m, \theta_m)$  est recherché, où  $\theta_m$  correspond à l’angle d’inclinaison.

La transformée de Hough a été testée sur des documents arabes imprimés. Mais, le problème majeur était la lenteur des calculs. Ainsi et afin de palier à ce problème, la réduction de points à traiter est nécessaire, et ce sans altérer la précision dans la détection de l’angle d’inclinaison recherché pour la totalité des points de l’image. L’utilisation de la méthode de Jain et Yu (Jain & al., 1996) ou de la méthode de Fletcher et Kasturi (Fletcher & al., 1988) donne de bons résultats pour les documents latins, mais elle n’est pas précise pour les documents arabes, car l’écriture arabe est cursive ainsi les points centraux des composantes connexes ne sont pas alignés sur la même ligne (exemple : pour le sous-mot “لك” et le sous-mot “في”, les centres de leurs composantes connexes respectives n’appartiennent pas à la même ligne), cela est dû au fait que les composantes connexes ne sont pas de tailles homogènes. Ce problème ne se pose pas pour les documents latins lorsque les caractères ne sont pas attachés.



**Figure 2.** (a) Texte arabe incliné, (b, d) Les points centraux des composantes connexes, (c) Texte latin incliné.

## 3. Caractéristiques de l’écriture arabe

L’écriture arabe est une écriture manuscrite, qui s’écrit de droite à gauche. En général les caractères arabes évoluent de façon ascendante ou descendante en prenant des allures minces ou épaisses. Chaque lettre arabe peut s’écrire de 2 à 4 formes différentes :

- Initiale : en début du mot (D).
- Médiane : au milieu du mot (M).
- Finale : à la fin du mot mais liée à une lettre précédente (Fl).
- Isolée : à la fin du mot sans être liée à la lettre précédente (Fs).

D	M	FI	Fs	D	M	FI	Fs
		أ	ا	ضد	ضد	ض	ض
ب	ب	ب	ب	ط	ط	ط	ط
ت	ت	ت	ت	ظ	ظ	ظ	ظ
ث	ث	ث	ث	ع	ع	ع	ع
ج	ج	ج	ج	غ	غ	غ	غ
ح	ح	ح	ح	ف	ف	ف	ف
خ	خ	خ	خ	ق	ق	ق	ق
		د	د	ك	ك	ك	ك
		ذ	ذ	ل	ل	ل	ل
		ر	ر	م	م	م	م
		ز	ز	ن	ن	ن	ن
س	س	س	س	هـ	هـ	هـ	هـ
ش	ش	ش	ش			و	و
ص	ص	ص	ص	ي	ي	ي	ي

Tableau 1. L'apparence des caractères de l'alphabet arabe.

Les caractères arabes sont toujours liés par des éléments d'épaisseur uniforme, le long d'un texte d'une même police et dont la position correspond au niveau de naissance de tout caractère; c'est le niveau de la ligne de jonction.

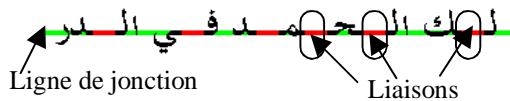


Figure 3. Représentation de la ligne de jonction

### 4. Détection de l'inclinaison en réduisant le nombre de points

La transformée de Hough permet de détecter avec exactitude l'angle d'inclinaison par contre le temps d'exécution est très important. La réduction du temps de traitement est indispensable ainsi le nombre de points doit être réduit. Pour cela, nous avons utilisé la connaissance à priori de l'écriture arabe cursive et qui consiste à utiliser les centres de liaisons inter-caractères dans la recherche de l'angle d'inclinaison, ce choix est justifié par le fait que ces liaisons :

- reposent sur la même ligne
- possèdent approximativement la même hauteur et la même largeur

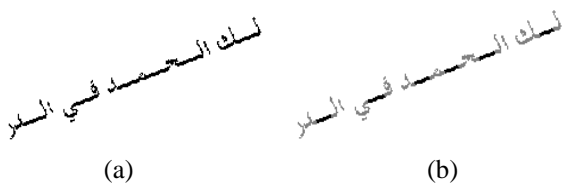


Figure 4. (a) Texte incliné, (b) Extraction des liaisons.

La détection de liaisons est effectuée suivant les étapes suivantes :

1. Extraction de composantes connexes

2. Filtrage de composantes connexes
3. Extraction de liaisons
4. Filtrage de liaisons
5. Application de la transformée de Hough sur les points centraux des liaisons.

#### 4.1. Calcul des composantes connexes

L'extraction des composantes connexes est réalisée au moyen d'un algorithme qui, sélectionne le premier pixel noir et recherche les pixels noirs qui lui sont voisins et à chaque fois les coordonnées du rectangle qui englobe ces pixels sont calculées. Pour se déplacer d'un pixel à un pixel voisin, nous avons utilisé un voisinage 8-connextité, c'est à dire qu'à partir d'un pixel de coordonnées (x, y), les déplacements sont effectués vers les huit pixels voisins dont les coordonnées sont : (x+1, y-1), (x-1, y+1), (x+1, y-1), (x+1, y+1), (x, y-1), (x, y+1), (x-1, y), (x+1, y).

Chaque composante connexe est identifiée par les coordonnées du coin supérieur gauche (x1, y1) et inférieur droit (x2, y2).

#### 4.2 Filtrage de composantes connexes

Cette étape consiste à sélectionner les composantes connexes contenant des sous mots; pour cela la hauteur et la largeur moyenne des composantes connexes sont calculées. Les composantes dont la hauteur et la largeur sont supérieures à la hauteur et à la largeur moyenne respectivement sont retenues, les autres composantes sont supprimées (Figure 5).

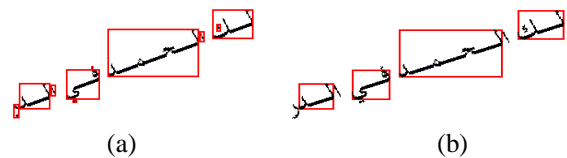
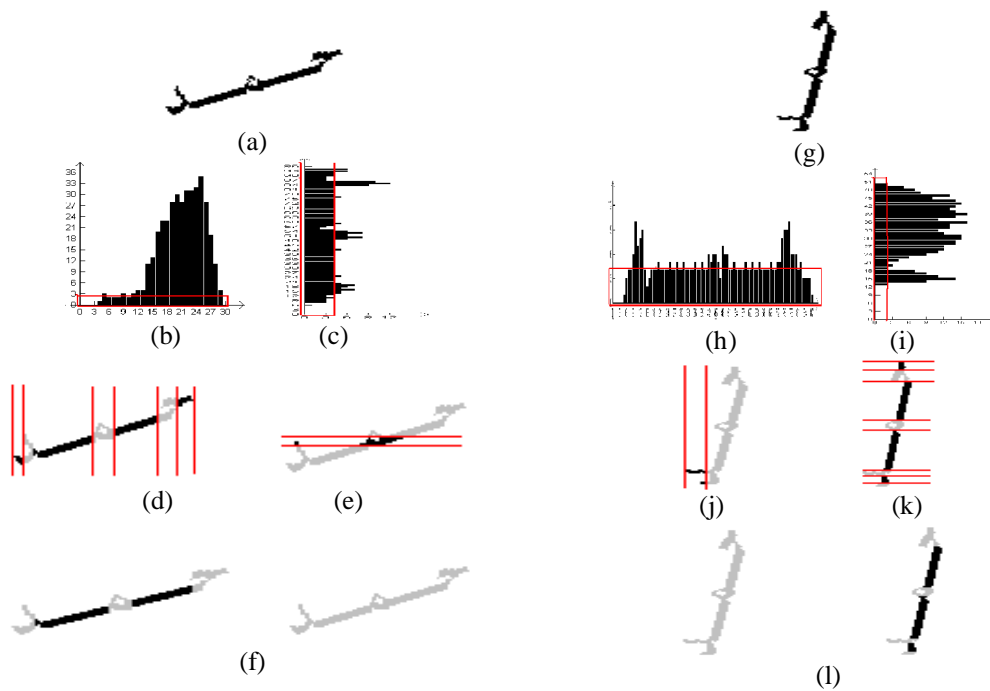


Figure 5. (a) Composantes connexes, (b) Filtrage de composantes connexes.

#### 4.3. Détection de liaisons

C'est l'étape la plus importante de notre approche, elle consiste à détecter les liaisons inter-caractères. Elle est effectuée en calculant les histogrammes horizontal et vertical pour chaque composante connexe qui contient un sous-mot. En général les liaisons sont caractérisées par une hauteur inférieure à la hauteur des caractères et leur fréquence d'apparition est très grande dans un sous-mot. Pour cela, la hauteur et la largeur les plus fréquentes sont calculées à partir des histogrammes. Ensuite, les liaisons sont extraites, elles correspondent aux zones dont la hauteur et la largeur est inférieure à la hauteur et à la largeur la plus fréquente dans l'histogramme vertical et horizontal respectivement.



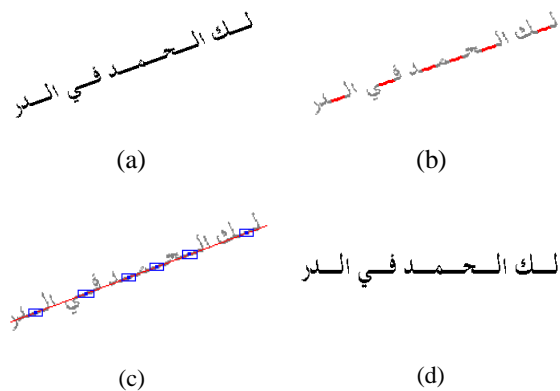
**Figure 6.** (a, g) Sous mot, (b, h) Histogramme vertical, (c, i) Histogramme horizontal, (d, e, j, k) Détection de liaisons, (f, l) Filtrage de liaisons.

$$HistoV [x] = \sum_{y=c.y1}^{c.y2} Pixel [x][y] \quad [1]$$

$$HistoH [y] = \sum_{x=c.x1}^{c.x2} Pixel [x][y] \quad [2]$$

'c' est la composante connexe d'un sous-mot.

$$Pixel[x][y] = \begin{cases} 0 & \text{si le pixel est noir} \\ 1 & \text{si le pixel est blanc} \end{cases} \quad [3]$$



**Figure 7.** (a) Texte incliné, (b) Détection de liaisons, (c) Les points centraux des liaisons, (d) Texte redressé.

Etant donné que l'utilisation de l'histogramme vertical ou horizontal dépend de l'orientation du document (Figure 6) qui doit être retrouvée. Nous avons calculé les deux histogrammes pour détecter les liaisons, ensuite, l'histogramme qui détecte plus de liaisons par rapport à l'autre sera gardé et l'autre sera éliminé.

#### 4.4. Filtrage de liaisons

Afin de sélectionner les parties de caractères qui peuvent être considérées comme liaisons, le prétraitement suivant est alors effectué, il consiste à :

- Eliminer les parties détectées au début ou à la fin d'une composante connexe car les liaisons se situent à l'intérieur des sous-mots (Figure 6(d) et Figure 6(k)).

- Eliminer les parties qui se chevauchent verticalement ou horizontalement (Figure 6(e) et Figure 5(j)).

#### 4.5. Détection de l'angle d'inclinaison

Une fois que les liaisons seront filtrées, la transformée de Hough sera appliquée sur les points centraux des liaisons en utilisant la formule suivante  $\rho = x * \cos\theta + y * \sin\theta$ , avec  $0^\circ \leq \theta \leq 180^\circ$ . Le vecteur accumulateur est examiné afin d'extraire la valeur maximale d'indices  $(\rho_m, \theta_m)$ , tel que  $\theta_m$  correspond à l'angle d'inclinaison.

Finalement, l'algorithme de détection de l'angle d'inclinaison se résume comme suit :

```

Début
Initialiser le tableau  $H(\rho, \theta)$  à 0
Calculer les composantes connexes
Filtrer les composantes connexes
Détecter les liaisons
Filtrer les liaisons
Pour chaque liaison faire
{
    Soit  $(x_c, y_c)$  le point central de la liaison
    Pour  $0 \leq \theta \leq \pi$ 
    {
        Calculer  $\rho = x_c * \cos\theta + y_c * \sin\theta$ 
        Incrémenter  $H(\rho, \theta)$ 
    }
}
Trouver les coordonnées  $(\rho_m, \theta_m)$  de la valeur
maximale du tableau  $H(\rho, \theta)$ 
 $\theta_m$  Correspond à l'angle d'inclinaison
Fin

```

## 5. Résultats expérimentaux

Afin de tester l'exactitude de l'algorithme de détection proposé dans cet article, nous l'avons appliqué sur cent images de documents arabes imprimés. Ces dernières ont été obtenues en scannant des livres, des lettres et des revues avec un scanner «Epson Expression 1600 Pro»; la résolution de numérisation était de 100, 150 et 200 dpi. Les images

sont sauvegardées sous format BMP (noir et blanc). Quelques images sont présentées dans la figure 8.

Les tests ont été établis en effectuant des rotations sur les documents scannés; l'angle de rotation testé lors des expériences varie entre  $0^\circ$  et  $180^\circ$ . Le déplacement dans l'espace de Hough était de un pour les axes  $\rho$  et  $\theta$  ( $\Delta\rho=1, \Delta\theta=1$ ).

Le temps de traitement est mesuré depuis le début de calcul des composantes connexes jusqu'à la fin de la détection de l'angle d'inclinaison. Les tests ont été effectués sur un PC doté d'un microprocesseur Pentium III 1GHz et une mémoire de 128 Mo sous Windows XP. L'application a été développée avec le C++ Builder 5.

Le tableau TAB. 2 illustre l'efficacité de notre algorithme sur un échantillon de 10 images choisies arbitrairement sur l'ensemble des 100 images traitées. Afin de comparer notre méthode avec la méthode originale de Hough, nous mentionnons dans ce tableau la taille de l'image, la résolution, l'angle estimé par chaque méthode, l'angle réel, le nombre de points traités, ainsi que le temps de calcul nécessaire pour le calcul de l'angle d'inclinaison. Comme nous pouvons le constater sur ces exemples, que notre algorithme permet de réduire le temps d'exécution d'une manière drastique; ainsi, le temps moyen de traitement de l'ensemble des 100 images est égale à 12 secondes si nous utilisons les points centraux des liaisons comparé à 30 minutes et 18 secondes dans la cas d'utilisation de tous les points de l'image. En outre, un autre constat de valeur est l'angle détecté par notre méthode qui est égal dans la plupart des cas à l'angle détecté par la transformée de Hough appliquée sur tous les points de l'image.

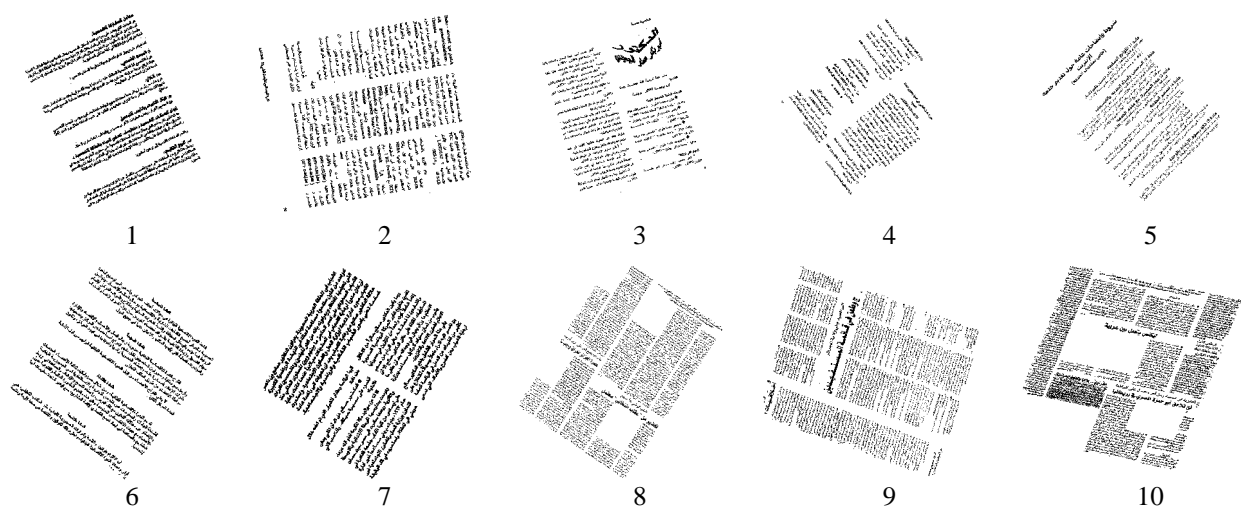


Figure 8. Exemple de quelques documents inclinés.

mage	Taille	Résolution	Angle estimé		Vrai angle	Nombre de points utilisés		Temps d'exécution	
			(1)	(2)		(1)	(2)	(1)	(2)
1	714*781	100	25	25	25	445	38627	8 s	4 mn 57 s
2	1237*1429	150	101	99	99	809	124168	7 s	35 mn 12 s
3	1801*2045	200	16	16	16	862	189136	11 s	48 mn 40 s
4	1577*1617	100	127	128	128	586	87130	6 s	22 mn 39 s
5	2149*2149	200	45	45	45	884	158332	12 s	44 mn 13 s
6	945*909	100	142	143	142	551	43859	5 s	7 mn 16 s
7	567*590	100	56	58	58	458	28895	3 s	3 mn 45 s
8	2709*2673	200	156	156	156	2025	344065	37 s	58 mn 16 s
9	2595*2286	200	72	72	72	4229	372100	58 s	1 h 30 mn 45 s
10	2014*1575	150	164	163	164	3220	279376	48 s	1 h 22 mn 47 s

**Tableau 2.** Tableau illustrant le temps de traitement.

- (1) La transformée de Hough appliquée sur les points centraux des liaisons
- (2) La transformée de Hough appliquée sur tous les points de l'image

Nous avons obtenu un taux de précision de 89% pour la détection exacte de l'angle de l'inclinaison, ce taux augmente jusqu'à 96% si nous considérons une marge d'erreur qui varie entre  $[-1^\circ, +1^\circ]$  (la précision de l'angle de l'inclinaison est égale au nombre de document dont l'angle d'inclinaison a été correctement détecté divisé par le nombre total de documents). Les imprécisions sont dues principalement aux liaisons détectées dans les caractères descendants (ص, ي, ر, ل, ن, ق) situés à la fin des sous-mots (Figure 9) et à la mauvaise qualité des images de documents.

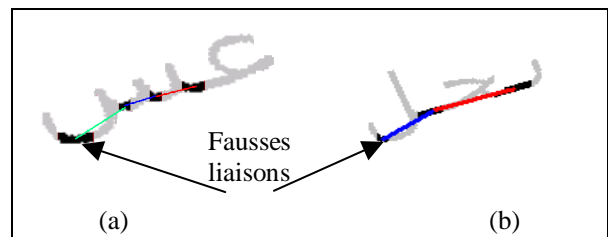
## 6. Conclusion

Dans cet article, nous avons présenté une nouvelle méthode pour la détection de l'inclinaison des documents arabes imprimés. Notre algorithme permet de diminuer le temps de traitement en réduisant le nombre de points à appliquer sur la transformée de Hough.

Cette réduction consiste à extraire les éléments reliant les caractères arabes et les modéliser par les centres de leur boîtes englobantes. Ce choix est justifié par le fait que ces éléments possèdent généralement la même largeur et la même hauteur et elles sont alignées sur la même droite. L'extraction de liaisons est effectuée selon plusieurs étapes : (1) Calcul des composantes connexes, (2) Sélection des composantes connexes qui contiennent des sous-mots, (3) Calcul de l'histogramme horizontal et vertical de chaque composante connexe afin d'extraire les liaisons et (4) Filtrage de liaisons. Une fois que les liaisons sont extraites la transformée de Hough est appliqué sur les points centraux de liaisons pour détecter l'angle d'inclinaison.

Les tests ont montré une bonne performance de l'algorithme. La précision dans l'estimation de l'angle d'inclinaison est très élevée et le temps d'exécution est acceptable. Comme perspective, nous proposons

de résoudre les problèmes rencontrés avec les lettres de les lettres de type "ص, ي, ر, ل, ن, ق" afin d'obtenir une meilleure précision. Pour cela, nous envisageons de rajouter une étape de vérification qui consiste à calculer l'orientation locale de chaque sous-mot qui contient plus de 4 liaisons; l'orientation est calculée par la méthode des k-plus proches voisins, ainsi les liaisons qui n'appartiennent pas à la direction trouvée seront éliminées (dans la figure 9(a), la dernière liaison est supprimée car elle n'appartient pas à la direction local du sous-mot).



**Figure 9.** Exemple de fausses liaisons

## 7. Références

- Amin A., Fischer S., Parkinson T., Shiu R., "Fast algorithm for skew detection", *SPIE Proceedings*, Vol. 2661, 29-30 Janvier 1996.
- Amin A., Shiu R., "Page Segmentation and Classification Utilizing Bottom-Up Approach", *International Journal of Image and Graphics*, Vol. 1, No. 2, 2001, pp. 345-361.
- Antonacopoulos A., "Local Skew Angle Estimation from Background Space in Text Regions", *Proceedings of the 4th International Conference on Document Analysis and Recognition*, 1997, pp. 684-688.
- Bagdanov A., Kanai J., "Projection Profile Based Skew Estimation Algorithm for JBIG Compressed Images", *In: Proceedings of the 4th International*

*Conference on Document Analysis and Recognition*, 1997, pp. 401-405.

Belaid A., Belaid Y., “*Reconnaissance des formes : méthodes et application*”, InterEdition 1992.

Bergler S., Khoury S., Suen B. C. Y., Waked B., “Skew Detection, Page Segmentation and Script Classification of Printed Document Images”, *IEEE International Conference on Systems, Man, and Cybernetics*, October 1998, pp. 4470-4475.

Bunke H., Wang P. S. P., “*Handbook of character recognition and document analysis*”, Edition World Scientific Publishing, 1997.

Fletcher L. A., Kasturi R., “A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images”, *IEEE Transactions on PAMI*, 1988, pp. 910-918.

Hull J. J., Taylor S. L., “*Document Analysis System IP*”, World Scientific Edition, 1998.

Jain A., Yu B., “A Robust and Fast Skew Detection Algorithm for Generic Documents”, *Pattern Recognition*, Volume 29 No. 10, October 1996, p. 1599-1629.

Kavallieratou E., Fakotakis N., Kokkinakis G., “Skew angle estimation for printed and handwritten documents using the Wigner-Ville distribution”, *Image and Vision Computing*, 2002, pp. 813-824.

Kwag H. K., Kim S. H., Jeong S. H., Lee G. S., “Efficient skew estimation and correction algorithm for document images”, *Image and Vision Computing*, 2002, pp. 25-35.

Le D. X., Thoma G., Weschle H., “Automated Page Orientation and Skew Angle Detection for Binary Document Images”, *Pattern Recognition*, Volume 27, Number 10, October 1994, pp. 1325-1344.

O’Gorman L., “The document Spectrum for page layout analysis”, *IEEE Transactions on Page Layout Analysis and Machine Intelligent*, 1993, p. 1162.1173.

Parker J. R., “Algorithms for image processing and computer vision”, Wiley, John & Sons, Incorporated, 1996.

Perantonis S. J., Gatos B., Papamarkos N., “Block decomposition and segmentation for fast Hough transform evaluation”, *Pattern Recognition*, 1999, pp. 811-824.

Safabakhsh R., Khadivi S. “Document Skew Detection Using Minimum-Area Bounding Rectangle”, *The International Conference on Information Technology: Coding and Computing*, March 27 - 29, 2000, pp. 253-258.

Yin P.Y., “Skew Detection and Block Classification of Printed Documents”, *Image and Vision Computing*, 2001, pp. 567-579.

Yu Z., Ma H., “An Enhanced Skew angle Estimation Technique for Binary Document Images”, *Fifth International Conference on Document Analysis and Recognition*, September 20-22, 1999, pp. 165-168.