

I – Introduction

L'oeil est l'organe de la vision. Il nous permet d'interagir avec notre environnement en captant et en codant ses éléments constitutifs. Percevoir un objet met en jeu des traitements élaborés qui décomposent l'image en fréquences et en orientations ; la vision humaine est donc un haut niveau de perception. La vision par ordinateur n'a pas l'ambition de reproduire fidèlement le fonctionnement du système visuel humain ; elle en modélise les principales caractéristiques pour permettre à la machine de réaliser des tâches de reconnaissance de manière automatisée. L'une de ces caractéristiques est la détection et l'extraction de contours d'objets. Elle a donné lieu à la mise au point de nombreuses techniques. La plupart d'entre elles utilise des opérateurs locaux de premier ordre (technique du gradient, Prewitt, Sobel) ou de second ordre (Laplacien) suivis d'une recherche de maxima locaux. Toutefois, ces techniques donnent des résultats insuffisants sur une image réelle où les changements d'intensité sont rarement nets et elles nécessitent une opération de seuillage pour éliminer le bruit. Les nouvelles approches se sont avérées plus performantes. Elles procèdent par optimisation de critères prenant en compte un modèle prédéfini de contours à détecter. On peut citer, par exemple, la méthode de Hueckel dont le principe est d'ajuster les paramètres d'un modèle de contour idéal de sorte qu'il corresponde au mieux aux données de l'image.

Dans notre dossier, nous nous intéressons plus particulièrement à la méthode des filtres de Gabor. Ceux-ci doivent leur nom à Dennis Gabor (1900-1979), le physicien anglais qui a inventé l'holographie. Ces filtres, dont le fonctionnement est proche des traitements visuels humains, ont l'avantage d'être paramétrables en fréquence, et en orientation. Nous les avons implémentés et appliqués à diverses images. Leur étude constituera la première partie de ce rapport.

Par ailleurs, une partie de notre travail concerne l'application des filtres de Gabor à la classification des écritures. Dans le domaine du traitement d'images, on peut distinguer trois types de techniques de classification : les techniques déterministes (symboliques), probabilistes (réseaux de neurones) et ensemblistes (morphologie mathématique). Parmi les nombreux projets de recherche qui ont été conduits dans ce domaine en France, nous pouvons citer le projet AGORA pour la Bibliothèque Humaniste Virtuelle, dont le principe est une classification de blocs pour la segmentation en zones de documents, et qui permet de rechercher des documents anciens en fonction de scénarii définis par l'utilisateur. La seconde partie de ce dossier portera sur une application possible des filtres de Gabor pour l'analyse et la classification des écritures manuscrites.

Enfin, nous discuterons les choix effectués ainsi que les résultats obtenus durant nos recherches.

II – Les filtres de Gabor : étude et application

II.1 – Fonction et Filtre de Gabor

Nous appelons *fonction de Gabor* l'association d'une courbe de Gauss et d'une sinusoïde orientée (voir *FIG. I.1*). En traitement d'images, nous travaillons dans le domaine spatial en dimension 2, ce qui nous permet d'écrire la fonction de Gabor de la manière suivante :

$$G(x, y, \theta, f) = e^{-\frac{1}{2} \left(\frac{x_{\theta}^2}{\sigma_x^2} + \frac{y_{\theta}^2}{\sigma_y^2} \right)} \cos(2\pi f x_{\theta})$$

$$\text{avec } x_{\theta} = x \cos \theta + y \sin \theta$$

$$\text{et } y_{\theta} = y \cos \theta - x \sin \theta$$

où θ est l'orientation de la sinusoïde, f sa fréquence et σ_x (respectivement σ_y) l'écart-type de la gaussienne selon l'axe des abscisses (resp. des ordonnées).

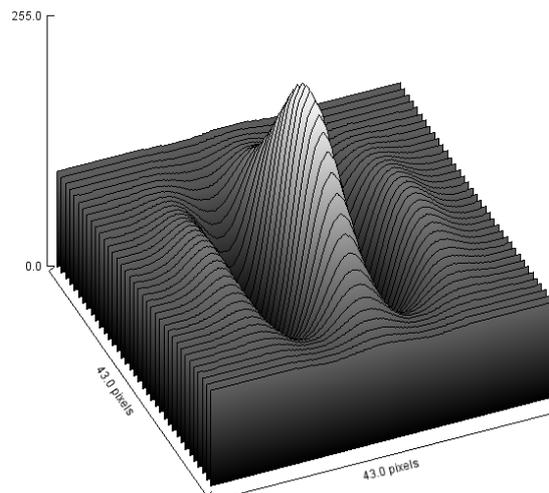


FIG. I.1 : représentation tridimensionnelle de la fonction de Gabor normalisée entre les valeurs 0 et 255

En appliquant cette fonction à un masque de convolution, on définit un filtre de convolution que nous appelons *filtre de Gabor* (voir *FIG. I.2*).

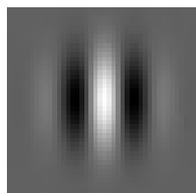


FIG. I.2 : masque du filtre de Gabor de rayon 21 pixels pour $\theta=0$, $f=\sqrt{2}/10$ et $\sigma_x=\sigma_y=7$

L'application d'un filtre de Gabor g de masque M de rayon r à une image I de largeur m et de hauteur n , se résume donc à la formule suivante :

$$g(I) = J = M * I$$

où J est une matrice de dimension m, n et pour $i, j \in \mathbb{N}$, $r \leq i < m - r$ et $r \leq j < n - r$:

$$J_{i,j} = \sum_{k=-r}^r \sum_{l=-r}^r M_{k,l} \times I_{(i-k),(j-l)}$$

$$= \sum_{k=-r}^r \sum_{l=-r}^r G(k, l, \theta, f) \times I_{(i-k),(j-l)}$$

Comme nous allons le voir, les filtres de Gabor permettent d'isoler les contours d'une image d'orientation perpendiculaire à θ et répondant à une certaine épaisseur, qui dépend de f . Ceci justifie le fait que pour détecter l'ensemble des contours d'une image, on lui applique généralement un ensemble de filtres de Gabor que nous appelons *banc*.

Nous avons donc implémenté les bancs de filtres de Gabor, afin de les utiliser sur des images diverses. Lors de l'exécution du programme, il est demandé de définir tous les paramètres utiles au calcul des fonctions de Gabor et à l'application du banc (voir *FIG. I.3*).

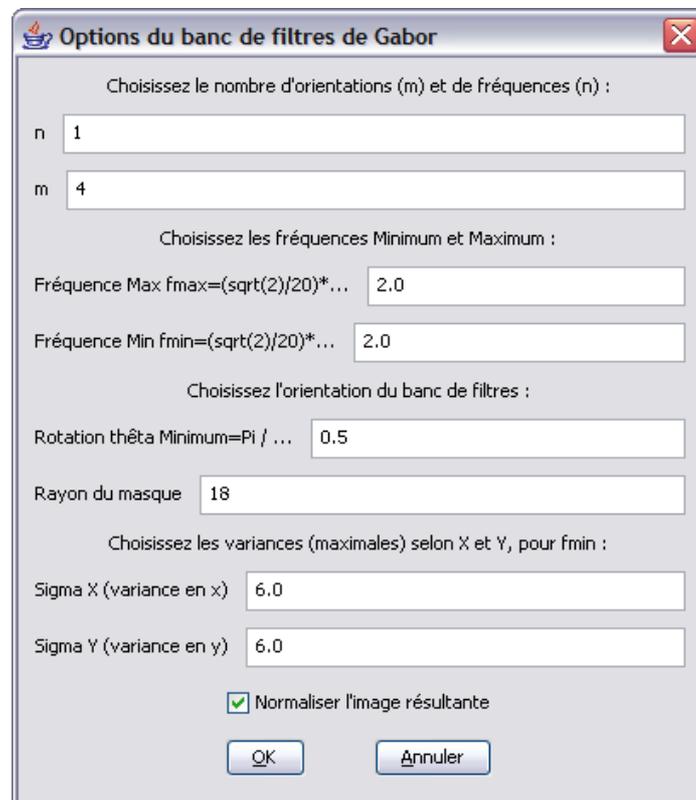
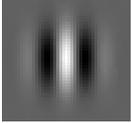
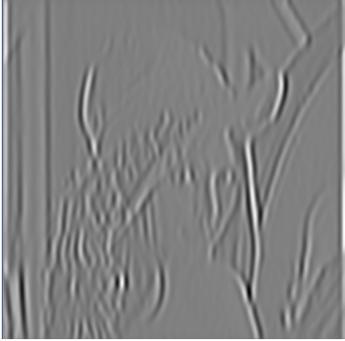


FIG. I.3 : saisie des paramètres d'un banc de filtre de Gabor

Le tableau suivant (*TABLEAU 1*) consigne divers exemples d'utilisation des filtres de Gabor sur une même image, afin d'expliciter l'influence des différents paramètres.

		<p>Orientation :</p> <p>L'application d'un seul filtre fait ressortir les contours orientés perpendiculairement par rapport à son orientation.</p>
		<p>Orientations multiples :</p> <p>L'application d'un banc à plusieurs orientations permet de détecter des contours plus nombreux.</p>
		<p>Écart-types :</p> <p>Si l'on fait varier σ_x de manière à ce qu'il soit inférieur à σ_y, les filtres deviennent moins sensibles aux contours et le banc se comporte (dans un état limite) comme un simple flou gaussien.</p> <p>Si l'on applique la variation inverse, le résultat, légèrement plus net, ne présentera pas de grandes différences avec un filtrage symétrique.</p>

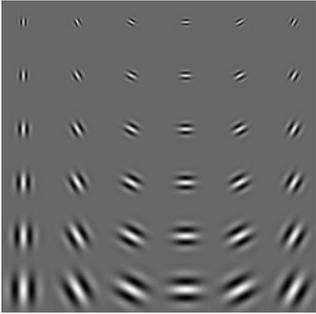
		<p>Fréquence : Si on utilise plusieurs fréquences, le tracé devient beaucoup plus net, le filtre est sensible à de plus nombreux contours en termes d'épaisseur.</p>
---	--	---

TABLEAU 1 : Influence des paramètres du filtre de Gabor

Ainsi l'association d'un ensemble de filtres de Gabor permet de couvrir plus largement l'espace fréquentiel et de détecter un plus grand nombre d'orientations, donc d'extraire tous les contours de l'image.

Le résultat d'un banc de filtres étant la moyenne des résultats des filtres de ce banc, nous pouvons tirer partie de la distributivité du produit de convolution afin d'alléger le traitement.

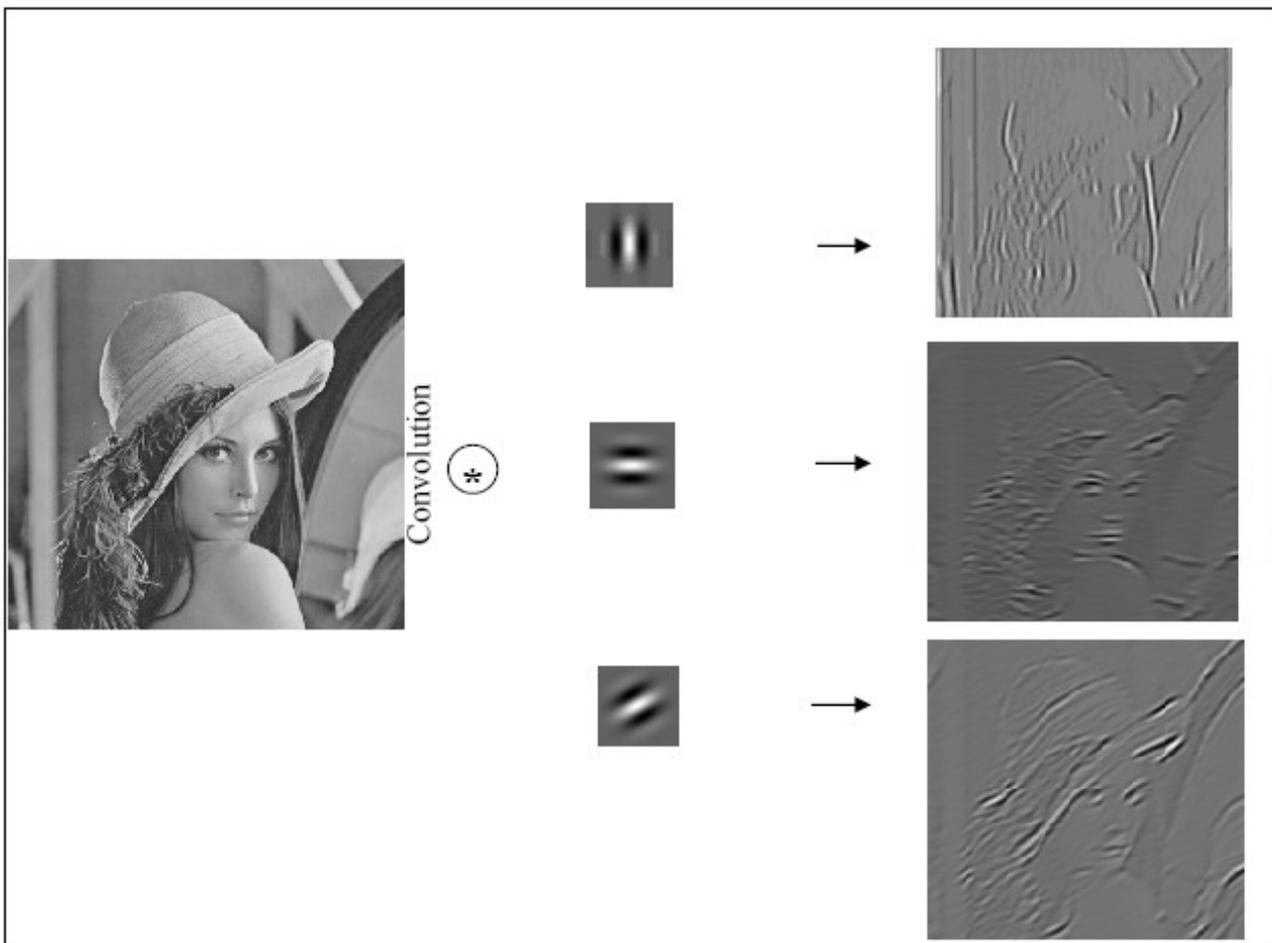


FIG I.4 : Convolution d'une image par un banc de filtres de Gabor

II.2 – Application à des images variées

Nous allons voir plus précisément dans cette partie l'application des filtres de Gabor à diverses images avec différents paramètres, afin de bien cerner leur fonctionnement.

Sur cette image (*FIG. II.2*), nous avons appliqué un banc de filtres à 4 directions (*FIG. II.1*).



FIG. II.1 : Banc de 4 filtres à fréquence fixe

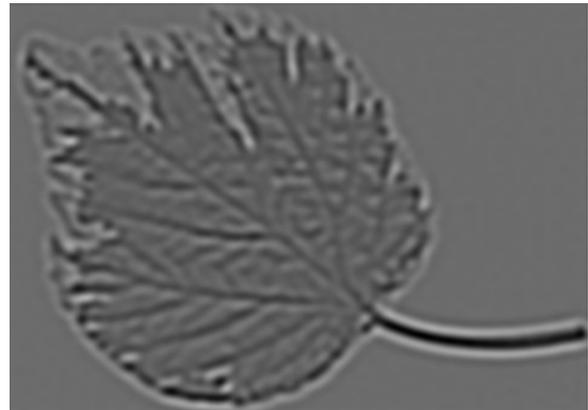


FIG. II.2 et II.3 : Image originale et résultat du traitement par le banc de la FIG. II.1

On constate que les contours sont bien identifiés, mais qu'ils ne sont pas vraiment précis. En effet, on remarque que l'image traitée devient un peu floue suite à l'application des masques.

En augmentant simplement le nombre de fréquences, on obtient plus de détails, comme on peut le voir sur la *FIG. II.5*. De ce fait, les contours deviennent plus précis que précédemment.

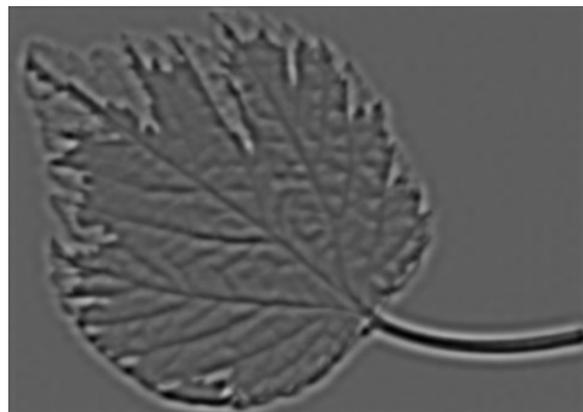
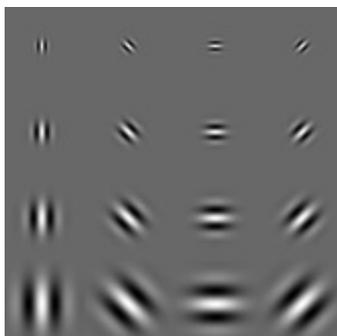


FIG. II.4 et II.5 : Banc de 8 filtres de Gabor, application à l'image de la FIG II.2

Ainsi, l'association d'un certain nombre d'orientations et de fréquences tout en tenant compte de l'écart type (afin que le filtre ne soit pas déformé), permet d'obtenir des résultats très fins (voir *FIG. II.6, II.7 et II.8*).

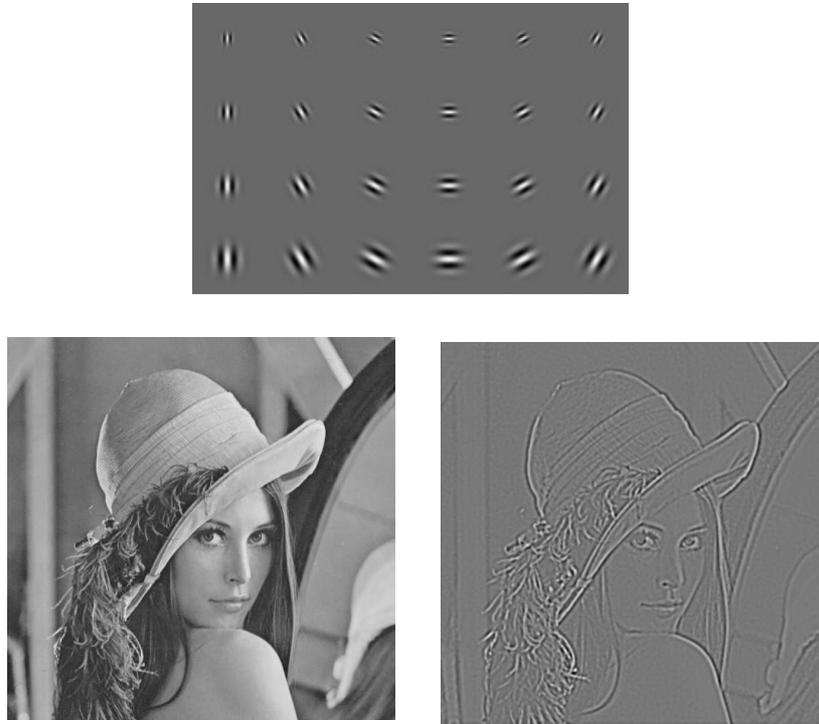


FIG. II.6, II.7 et II.8 : Banc de 24 filtres, image originale, résultat du filtrage

II.3 – Application à la classification des écritures

Dans cette partie, nous allons développer l'utilisation des filtres de Gabor dans l'analyse et la classification des écritures manuscrites. Comme nous l'avons vu précédemment, les filtres de Gabor sont paramétrables en fréquence ainsi qu'en orientation. Leur utilisation permet d'extraire les contours des images pour en caractériser la texture. Il est tout à fait possible d'utiliser ces caractéristiques afin d'obtenir une **carte de contours orientés** d'une image de document manuscrit, et de définir une **rose des directions** pour décrire sa texture.

II.3.1 - Obtention des cartes de contours orientés

Nous appelons carte des contours orientés, une matrice caractérisant des contours d'objets par opposition avec l'arrière-plan de l'image, tout en les différenciant selon leur orientation. Dans la pratique, notre carte des contours orientés est composée de pixels blancs pour l'arrière-plan, et de pixels de couleurs différentes pour les contours, chaque couleur correspondant à une orientation particulière.

Dans le cas des images de documents manuscrits, nous appliquons 4 (*respectivement* 6) filtres de Gabor, correspondant aux orientations 0° , 45° , 90° et 135° (*resp.* 0° , 30° , 60° , 90° , 120° et 150°). Contrairement aux traitements décrits précédemment, les filtres ne sont pas appliqués en banc, mais successivement, dans le but de pouvoir traiter séparément les différentes cartes de Gabor que nous obtenons.

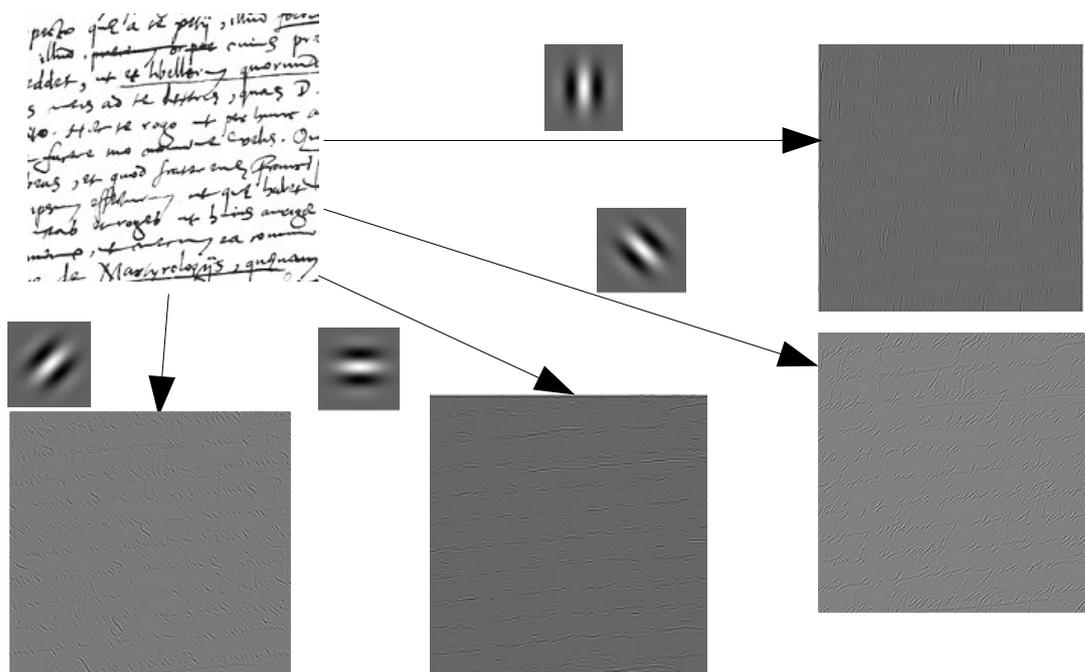


FIG. III.1 : Application de 4 filtres de Gabor à une image de document manuscrit

Les cartes orientées devant décrire les contours des écritures, nous allons uniquement garder les composantes les plus claires des cartes de Gabor, c'est-à-dire les pixels correspondant aux contours extérieurs des objets. Pour ce faire, et pour limiter les effets liés au bruit, nous utilisons un algorithme de binarisation à deux seuils (un seuil haut et un seuil bas), fonctionnant de la manière suivante :

- Si le pixel observé est plus clair que le seuil haut, il s'agit d'un point de contour,
- Si le pixel est plus foncé que le seuil bas, il n'appartient pas à un contour,
- Si le pixel est entre les deux seuils, on détermine son appartenance ou non à un contour en vérifiant la présence de pixels supérieurs au seuil haut dans son voisinage.

Enfin, nous superposons les 4 cartes binaires obtenues en attribuant à chacune une couleur particulière. Il arrive qu'un même pixel aie été détecté simultanément par plusieurs filtres de Gabor. Dans ce cas, nous ne prenons en compte que la direction pour laquelle ce pixel a été le plus réactif. Ce dernier traitement résulte en l'obtention d'une carte des contours orientés.

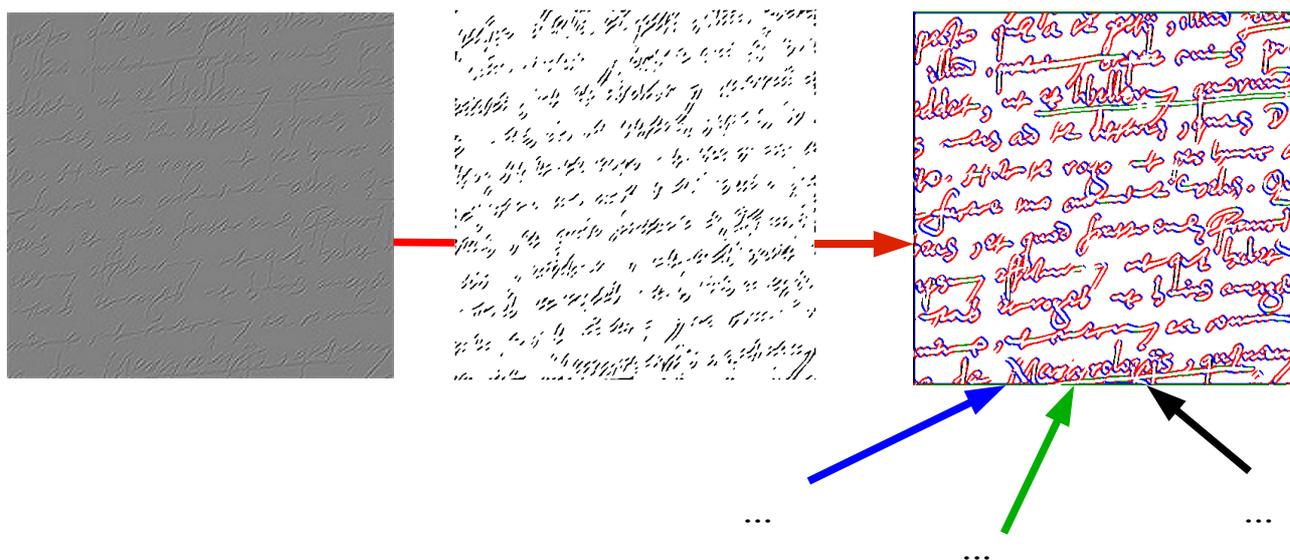


FIG. III.2 : Binarisation et affectation d'une couleur (ici le rouge) à chaque carte de Gabor, puis superposition de celles-ci.

II.3.2 - Calcul et utilisation des roses des directions

Une rose des directions est un diagramme faisant figurer les proportions relatives entre les composantes directionnelles d'un objet. Par un souci de facilitation de langage, nous appellerons aussi bien « rose des directions » le vecteur à 4 (resp. 6) dimensions dont les coordonnées sont les proportions des composantes de l'image, que le diagramme modélisé par ce vecteur.

Plus simplement, la rose des directions décrit les pourcentages de pixels de couleur rouge, bleue, verte, et noire dans le dessin de la carte des contours orientés. Dans la pratique, nous calculons cette rose directement depuis les 4 (*resp.* 6) cartes de Gabor. Nous expliquerons plus avant les raisons de ce choix dans la discussion de nos choix d'implémentation.

Lors de la construction de la rose, chaque pixel vote pour chaque direction, avec un poids normalisé proportionnel à sa valeur sur les cartes de Gabor correspondantes. Ainsi, les pixels les plus réactifs à une direction particulière favoriseront celle-ci dans leur vote par rapport aux pixels les moins réactifs. Afin de ne garder que les contours extérieurs, nous ne prenons ce vote en compte que si la valeur du pixel sur la carte de Gabor est supérieure à la médiane.

Nous obtenons de cette manière les 4 (*resp.* 6) coordonnées de notre rose des directions : celles-ci sont de somme égale à 100.

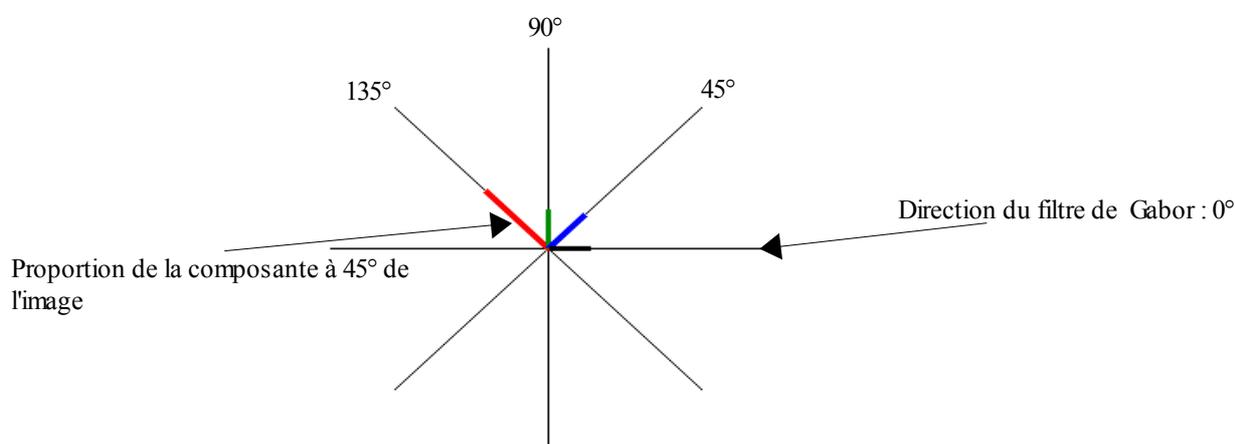


FIG III.3 : Rose des directions pour le vecteur (16,3 ; 24,8 ; 16,9 ; 42)

Les roses des directions sont étroitement liées aux paramètres des filtres de Gabor utilisés (la direction, clairement, et la fréquence comme nous le verrons plus loin), mais, à paramètres constants, elles possèdent deux avantages :

- Elles sont indépendantes de l'échelle des objets observés. Deux objets de même forme et de tailles différentes auront la même rose des directions.
- Sur une même écriture, elles sont constantes à condition que l'échantillon observé soit suffisamment représentatif du style du scripteur.

Ces deux caractéristiques de la rose des directions nous permettent de dire qu'à une texture particulière va correspondre une rose des directions précise, et nous fournissent un outil de comparaison entre les écritures.

II.3.3 - Comparaison et classification des écritures

Pour ce qui concerne la comparaison et la classification des écritures, nous travaillons essentiellement sur le vecteur « rose des directions ». En considérant ce vecteur comme un point dans un espace affine de dimension 4 (*resp.* 6), nous obtenons directement une échelle de comparaison entre deux écritures, la distance euclidienne :

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$$

En effet : plus les roses de deux images sont proches, plus les textures de ces images sont semblables ; par conséquent, les lettres définissant ces textures se ressembleront probablement.

Par ailleurs, la somme des coordonnées des roses des directions est égale à 100, ce qui s'écrit :

$$a_1 + a_2 + \dots + a_n = 100$$

Cette équation définit un hyperplan de \mathbb{R}^n . Le fait que ces coordonnées soient des pourcentages (donc des valeurs comprises entre 0 et 100) réduit l'hyperplan à un sous-espace borné, que nous notons $r(\mathbb{R}^n)$, l'ensemble des roses des directions de \mathbb{R}^n .

On peut montrer que la distance euclidienne induite sur ce sous-espace est bornée et comprise entre 0 et $100\sqrt{2}$, ce qui veut dire que nous pouvons définir un *pourcentage de similarité* comme suit :

$$s_{x,y} = 100 - \frac{d(x,y)}{\sqrt{2}}$$

$$\text{où } x, y \in r(\mathbb{R}^n) \text{ et } s_{x,y} \in [0, 100]$$

Ce pourcentage nous permet d'exprimer de manière explicite et intuitive la ressemblance entre deux styles d'écriture. Deux images identiques du point de vue de la texture seront dites « similaires à 100% ».

Nous pouvons maintenant envisager deux utilisations de ce critère de comparaison : la reconnaissance du scripteur, et la classification des images selon le style d'écriture.

II.3.4 - Expérimentations

Nous avons tenté deux expériences distinctes à partir des traitements que nous venons de décrire, sur une base d'images de lettres manuscrites anciennes. Celles-ci, au nombre de 20, ont été *scindées* en trois (voire quatre) pour obtenir 64 images réparties en 4 séries, que nous nommons de la manière suivante : *[numéro de la série].[numéro du document]* .

Par exemple, « 1.167 » et « 2.167 » sont deux images distinctes provenant de la même lettre d'origine.

Nous avons appris à notre système les roses des directions de la première série d'images. Nos tests portent donc sur les 3 autres séries.

La première expérience consiste à soumettre une image au système en lui demandant de nous renvoyer l'image de la base la plus ressemblante. Le système va alors appliquer la chaîne de traitements nécessaire au calcul de la rose des directions de cette image, puis va chercher dans la base l'entrée minimisant la distance euclidienne (maximisant le pourcentage de similarité) et retourner cette dernière. Nous distinguons 3 types de résultats *a priori* :

- Exact : l'image retournée par le système provient du même document,
- Similaire : l'image retournée provient d'un document différent, mais présente des similarités en termes de formation de l'écriture (indépendamment de l'échelle),
- Incorrect : l'image retournée n'est ni exacte, ni similaire.

Cette expérience consiste clairement à identifier le scripteur. Nous verrons plus loin en quoi les résultats de celle-ci nous ont incité à tenter la seconde expérience, plus proche des capacités de notre système : la classification des images selon le style de l'écriture.

Pour effectuer cette dernière, nous avons complété les connaissances de notre système en lui apprenant le résultat d'une classification ascendante hiérarchique (CAH) sur les images de la première série, puis, en lui soumettant les images des autres séries, nous lui avons demandé de nous retourner la classe correspondante. Celle-ci est déterminée par la minimisation de la distance euclidienne entre l'image testée, et les points moyens (*barycentres équipondérés*) des classes connues du système. Pour cette seconde expérience nous avons distingué *a priori* 2 types de résultats :

- Exact : la classe retournée contient un échantillon de la même lettre que l'image testée,
- Inexact : la classe retournée ne contient pas un tel échantillon.

II.4 – Résultats

II.4.1 - Reconnaissance du scripteur

Le Tableau 2 consigne les résultats du test de reconnaissance du scripteur

Dimension	Exact	Similaire	Inexact
4	20,5 %	70,5 %	9 %
6	11 %	34 %	31 %

TABEAU 2

II.4.2 - Classification des écritures

Le Tableau 3 représente les résultats du test de classification des écritures, en 2 classes.

Dimension	Exact	Inexact
4	91 %	9 %
6	77 %	23 %

TABEAU 3

III – Discussion

III.1 – Choix d'implémentation

III.1.1 – Filtres et Convolution

Dans la partie II.1, nous évoquons la distributivité du produit de convolution. Celle-ci est effectivement exploitée dans l'implémentation des bancs de filtres de Gabor, ce qui permet de réduire considérablement le temps de traitement lors de leur utilisation.

La convolution est connue pour être un opérateur lourd et complexe algorithmiquement parlant. En effet, si nous convoluons une matrice à n lignes et p colonnes par un masque carré de rayon r , cela revient à appliquer un nombre de multiplications de l'ordre de $np \times 2r^2$ ce qui, lorsque n et p sont les dimensions d'une image numérisée, correspond à un temps de traitement de l'ordre de 30 secondes à 1 minute sur une machine moyennement puissante. Dans le cas où nous appliquons un banc de filtres à f fréquences et o orientations, cela appesantit encore la complexité de calcul d'un facteur $f.o$. En considérant un traitement à 8 fréquences et 8 orientations, on obtient un calcul qui peut durer plus d'une heure, d'où la nécessité de rentabiliser l'utilisation du produit de convolution.

Il existe effectivement un moyen de n'effectuer qu'une seule convolution, quel que soit le nombre de filtres du banc : factoriser les masques. Plutôt que de calculer la moyenne des résultats des

convolutions de l'image par des masques successifs, nous convoluons cette image par la moyenne des masques ; ce qui, dans notre exemple où $f = o = 8$, aboutit à un résultat identique, et un traitement 64 fois plus rapide.

III.1.2 – A propos de la fréquence

Dans notre implémentation, nous nous sommes aperçus qu'en termes de fréquence, l'affichage des masques ne correspondait que rarement à ce que nous souhaitions. Nous en avons isolé la cause :

Lorsque nous affichons une fonction de Gabor à l'écran, nous la discrétisons, de manière à l'appliquer à des entiers naturels (l'espace défini par les pixels d'une image). Ce faisant, la valeur des pixels est approchée au plus proche entier, ce qui, avec une fréquence « élevée », résulte en une image uniforme, en lieu et place d'une alternance de raies blanches et noires. Afin de pallier à ce problème, nous utilisons des fréquences basses.

Afin de simplifier le paramétrage de nos filtres, nous avons décidé de désigner la fréquence par des nombres simples (2, 4, ... ,16) en définissant une fréquence échelonnée $f_e = (\sqrt{2}/20) f$.

Ceci nous permet donc d'afficher des masques de Gabor sans biais dûs à la discrétisation, en utilisant une échelle simple à paramétrer, tout en sachant quelle fréquence réelle nous utilisons.

À propos de paramétrage, nous pouvons remarquer que l'écart-type de la gaussienne doit être, comme pour un filtre gaussien, inférieur à 1/3 du rayon du masque afin de respecter la dynamique de la fonction de Gabor : après 3 écarts-types, la gaussienne devient négligeable. Dans la plupart de nos traitements, nous la lions à la fréquence échelonnée f_e par la formule :

$$\sigma = \frac{16}{f_e} \leq \frac{r}{3}$$

Ceci nous procure un masque où nous observons une bande blanche centrale (le mode) cernée par deux bandes noires (les deux gouffres de la *FIG I.I*), qui vont tendre continûment vers 0 (ou vers la médiane si le masque est normalisé en niveaux de gris). Bien entendu, ces résultats n'ont qu'une valeur empirique, mais ils facilitent l'utilisation de notre programme et celle des filtres de Gabor en général.

Un autre aspect très important de la fréquence que nous devons développer ici est son influence sur la carte des contours orientés et sur la rose des directions. Le lecteur pourra remarquer que les deux expériences que nous tentons dans la partie II.3 se font à fréquence constante, nous allons justifier cet état de fait.

Les figures IV.1 à IV.5 illustrent la variabilité de la carte des contours orientés en fonction de la fréquence utilisée pour l'obtenir.



FIG. VI.1 : image originale

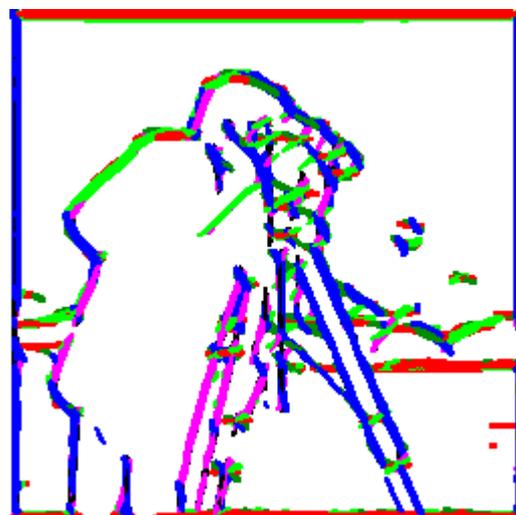
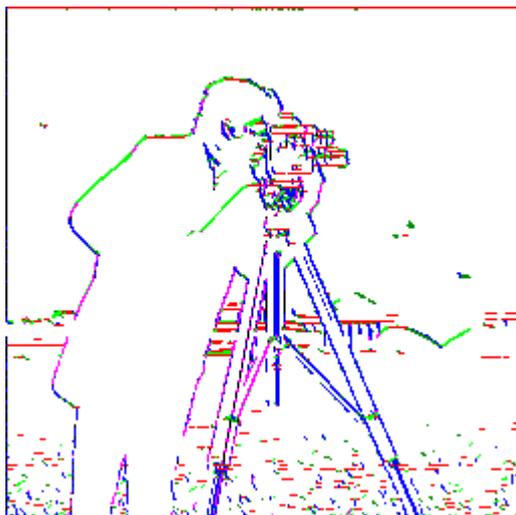


FIG. IV.2, IV.3, IV.4 et IV.5 : Cartes des contours orientés à 6 directions pour des fréquences échelonnées respectives $f_e = 16, 8, 4$ et 2

La première remarque que nous pouvons faire est que la fréquence influe sur l'épaisseur du trait de la carte des contours orientés. En effet, plus la fréquence est basse, plus les contours de la carte seront grossiers. Il en découle que les contours les plus fins sont négligés à très basse fréquence, et que les plus grossiers sont incomplets à très haute fréquence.

Mais l'influence la plus déterminante sur les traitements ultérieurs est très certainement le fait que, pour deux fréquences différentes, un même contour n'est pas nécessairement le plus réactif pour le même filtre. On constate en effet sur les figures précédentes que les contours n'ont pas forcément la même couleur d'une carte à l'autre. Ceci se traduit sur la rose des directions par des coordonnées n'ayant rien en commun entre elles d'une fréquence à l'autre.

Étant donné que, pour nos expériences, les traitements appliqués à chaque image doivent être identiques pour que les résultats aient une valeur statistique quelconque, nous pouvons envisager deux solutions simples à ce problème de variabilité, et de choix de la fréquence.

La première solution est d'appliquer plusieurs fréquences aux images de manuscrits de la même façon que nous leur appliquons plusieurs directions, puis de moyenniser les résultats. Cette voie entraîne un temps de calcul beaucoup plus long (celui-ci se trouve multiplié par le nombre de fréquences) et ne semble pas pertinente pour nos analyses puisque, en général dans une écriture, l'épaisseur des traits peut être considérée comme constante.

La seconde solution est de déterminer, manuellement ou de manière automatisée, la fréquence la plus adaptée à l'ensemble des documents étudiés : celle pour laquelle le plus grand nombre de contours sont lisibles sur les cartes des contours orientés.

Nous avons donc choisi la deuxième solution, car elle nous semblait la plus appropriée dans le cadre de l'analyse de documents manuscrits. Ainsi, les roses des directions de tous les documents (à traiter ou appris par le système) sont calculées avec une fréquence unique, que nous avons déterminé de manière empirique : $f_i = 8\sqrt{2}/20$.

III.1.3 – Roses des directions

Plusieurs solutions étaient possibles pour la création de nos roses des directions. Par exemple, nous aurions pu simplement compter le nombre de pixels de couleur sur la carte des contours orientés puis normaliser les résultats de 0 à 100 ou leur donner un poids proportionnel à leur valeur sur la carte de Gabor correspondant à leur couleur. Si nous avions utilisé l'une ou l'autre de ces deux solutions, nous aurions négligé une bonne partie des informations fournies par les cartes de Gabor, à cause des seuils de binarisation, ce qui justifie le fait que nous les calculions directement à partir de ces cartes de Gabor.

Par ailleurs, il a été précisé que nous ne retenons que les pixels supérieurs à la médiane d'une carte de Gabor donnée, plutôt que l'ensemble de la carte. Ce choix est dû à la forme de l'histogramme cumulé d'une carte de Gabor.

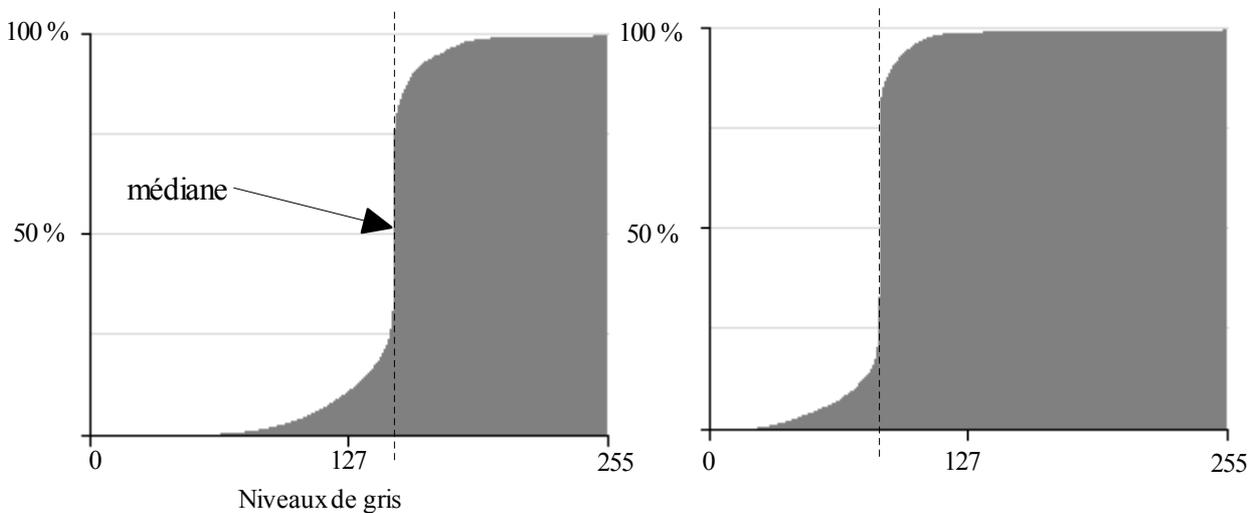


FIG. IV.6 et IV.7: Exemples d'histogrammes cumulés d'une carte de Gabor ayant détecté une majorité de contours intérieurs (à gauche), et extérieurs (à droite).

L'histogramme d'une carte de Gabor a toujours la forme des FIG. IV.6 et IV.7, pour lesquelles varie la médiane (ici le point d'inflexion). Cette médiane dépend des contours qui ont été les plus réactifs au filtre de Gabor. Si les contours intérieurs ont mieux réagi, (FIG. IV.6), la médiane est décalée vers la droite. Dans le cas contraire (FIG. IV.7), elle est décalée vers la gauche.

On remarquera que les courbures des « semi-histogrammes » à gauche et à droite de la médiane sont plus ou moins larges selon la position de celle-ci. Ces courbures caractérisent en fait la netteté avec laquelle les contours ont été détectés.

Si, lors du calcul de la rose, nous voulions utiliser les contours intérieurs et extérieurs de l'image, nous devrions replier la carte de Gabor sur elle-même, en prenant appui sur la médiane, afin d'obtenir des votes normalisés positifs. L'histogramme d'une carte repliée aurait donc l'allure de la partie droite de ceux que nous observons mais avec un rayon de courbure plus large (pour la figure 7) ou plus étroit (pour la figure 6). En effet, le rayon de courbure obtenu serait une moyenne des deux rayons de l'histogramme de base.

Par suite, les contours les plus réactifs seraient moins bien différenciés des autres.

Nous avons remarqué que cet effet se traduit par un *aplanissement* de la rose des directions : les valeurs des coordonnées se resserrent entre elles. À cause de cet aplanissement, les roses deviennent moins « caractéristiques » des images. De ce fait, nous avons estimé ne pas perdre d'information, et mieux respecter la netteté avec lesquels les contours sont détectés, en ne tenant compte que des contours extérieurs.

III.1.4 – Gestion des effets de bord

Comme tout filtre de convolution, le filtre de Gabor induit des effets de bord au calcul du résultat. Cela est dû au fait que nous ne pouvons pas centrer le masque sur les pixels du bord de l'image sans que celui-ci ne dépasse. Pour pallier à ce problème, une solution courante est de ne traiter que les pixels de l'image qui se trouvent à une distance supérieure au rayon r du masque, par rapport aux bords.

Or, il se peut que la fréquence du filtre que nous désirons appliquer soit assez basse, ce qui donne un rayon très large pour le masque. Si l'on appliquait cette solution, la quantité de pixels traités dans l'image pourrait être trop petite. Nous avons donc tenté de modifier l'image afin qu'elle soit entourée par un cadre d'épaisseur r . L'image renvoyée par le filtre est alors de même taille que l'image de départ, et l'on peut s'apercevoir que les effets de bords touchent un contour de l'image d'épaisseur inférieure à r : nous perdons moins d'information.

Il peut être utile de faire une autre précision à propos des effets de bord : lorsque nous calculons une rose des directions, nous en tenons compte en prenant les pixels se situant à une distance $r/2$ du contour. Ceci nous permet de ne pas fausser les coordonnées. Pour vérifier cette affirmation, il suffit d'afficher une rose des directions sur une image unie (avec un niveau de gris quelconque) et vérifier qu'elle est égale au vecteur nul.

III.2 – Discussion des résultats

Nous allons maintenant apporter un point de vue critique sur les résultats des expériences décrites dans le paragraphe II.3.4.

Tout d'abord, nous allons tenter de les interpréter. Nous voyons clairement que le système que nous avons conçu ne permet pas d'identifier le scripteur d'un document. En effet, nous n'obtenons que 20,5 % de réponses exactes dans le meilleur des cas, ce qui prouve qu'une application de ce système à la reconnaissance du scripteur n'est absolument pas envisageable. En revanche, nous pouvons remarquer la prépondérance de réponses « similaires » pour ce même test. Dans 70% des cas en dimension 4 (34% en dimension 6), le système confond l'écriture que nous lui soumettons avec une autre écriture de texture semblable. Nous pourrions imaginer obtenir les mêmes résultats en demandant à un sujet humain d'effectuer la même reconnaissance, en lui présentant les images à une distance telle qu'il ne puisse pas en déchiffrer les mots. Nous en déduisons que notre système semble plus à même de caractériser un style d'écriture, que de reconnaître la personne qui en est l'auteur. C'est la raison pour laquelle nous avons procédé à la deuxième expérience : la classification des écritures.

Nous avons effectué celle-ci sur deux classes d'apprentissage obtenues en soumettant les roses de notre base d'apprentissage à un logiciel de statistique, et en effectuant une CAH de ces données, en prenant comme critère le lien moyen dans une métrique euclidienne. Le choix de ce critère d'agrégation semble justifié au vu du type de comparaison que nous effectuons, mais on peut remarquer que la grande majorité des autres critères conduit aux mêmes classes (on peut donc les considérer comme stables). Ces deux classes ne varient que d'un seul individu entre les deux dimensions utilisées et sont intuitivement séparables. La plus peuplée des deux correspond à une écriture que nous qualifierons de « régulière », souvent composée de lettres arrondies et bien formées, alors que la seconde écriture est plus compacte, et se compose de lettres irrégulières.

Pour ce second test, les résultats sont conformes à ce que nous attendions de notre système (seulement 9 % d'erreur en dimension 4 et 23 % en dimension 6) mais on peut s'étonner de voir que le système, supposé plus précis avec 6 dimensions, aie toujours des résultats plus faibles qu'avec 4. C'est ce que nous allons tenter d'expliquer.

Le choix de 6 dimensions est une réponse que nous pensions adaptée au fait que la plupart des écritures sont penchées d'environ 60° , plutôt que 45° , et nous supposons que le système détecterait des détails plus précisément en tenant compte de cette orientation. Au vu des résultats, il apparaît que c'est l'inverse qui se produit. Ceci peut s'expliquer en considérant que, par la nature même des écritures manuscrites, la rose des directions va être déséquilibrée vers la coordonnée correspondant à 60° .

Pour simplifier, on admet que ces contours vont prendre 50% de la rose sur une coordonnée particulière. Sur une rose à 6 directions, la proportion restante doit se partager entre les 5 autres coordonnées, contre 3 en dimension 4.

Inévitablement, la variance des coordonnées sera beaucoup plus élevée en dimension 4.

Ceci aboutit à un nuage de points plus compact, présentant moins d'inertie pour 6 directions que pour 4. Cette différence d'inertie a un impact direct sur les distances entre les roses des directions (plus le nuage est éclaté, plus grandes seront les distances), et c'est ce phénomène qui serait à l'origine de cette faiblesse des roses à 6 dimensions.

IV – Conclusion générale

Les filtres de Gabor sont des opérateurs particulièrement commodes pour l'extraction et la détection de contours. En effet, nous sommes capables avec leur aide d'isoler dans une image des composantes très variées, qui vont de gros objets clairement définis à de fins détails d'orientation particulière, en changeant simplement deux paramètres : la fréquence et l'orientation. Ces deux éléments indissociables sont les deux caractéristiques nécessaires et suffisantes à la description d'une ligne de contour : son épaisseur et sa direction. Il a été montré que le système visuel humain procédait de manière analogue pour la détection de contours. En ceci, nous ne saurions classer la méthode des filtres de Gabor dans les « anciennes » ou les « nouvelles » techniques, nous devrions plutôt la qualifier de technique *naturelle*.

En ce qui concerne l'analyse de l'écriture, nous sommes encore, évidemment, loin de reproduire le modèle humain. Modéliser la vision ne suffit pas, encore faut-il reproduire les mécanismes qui entrent en jeu lorsque nous observons et jugeons le style d'une écriture manuscrite pour en identifier l'auteur. Certes, la caractérisation par approche « texture » nous fournit un outil de classification relativement efficace (du moins dans le cadre de nos expériences), mais cet outil est indiscutablement perfectible, et, associé à d'autres techniques existantes, pourrait à coup sûr devenir plus précis.

Les progrès de l'informatique et l'automatisation soulagent l'homme de nombreuses tâches, parfois de manière si complète et si efficace que nous en avons oublié comment les effectuer autrement. On peut entrevoir de nombreuses actions que nous pourrions déléguer à la machine dans le domaine de l'analyse de l'écriture, et plus généralement de la vision par ordinateur, mais, sans nul doute, l'homme devra encore compter sur la complexité de son système visuel pendant longtemps.