

Reconnaissance Matricielle de Caractères manuscrits

Introduction

Cette partie traite d'un cas simple de reconnaissance de formes (RdF). Il constitue un apport supplémentaire pour la confrontation de plusieurs scores de reconnaissance (cf. Chapitre III § 3. de la thèse sur la reconnaissance de l'écriture manuscrite) pour des caractères isolés.

La Reconnaissance Matricielle des Caractères est une procédure de reconnaissance élémentaire car elle ne comporte pas l'étape d'extraction de caractéristiques et de primitives autres que celles faisant appel à la notion de pixel. Tout au plus allons nous considérer un ensemble irréductible de pixels, répartis dans une matrice rectangulaire pouvant représenter l'ensemble des caractères à reconnaître.

Cette méthode est particulièrement adaptée aux caractères majuscules imprimés ou manuscrits (caractères bâton) s'ils sont tracés avec application.

En supposant que ces pixels soient invariants entre les caractères candidats à la reconnaissance et l'ensemble de référence (lettres de 'A' à 'Z'), nous allons procéder à un ajustement matriciel (recadrage des caractères) afin de faire correspondre les tailles des caractères à comparer.

1. Ajustement matriciel

On cherche à faire correspondre une matrice de taille quelconque, représentant un caractère candidat, avec une matrice de la taille bien définie pour laquelle on a prévu l'apprentissage. En effet, la correspondance des pixels d'une matrice à une autre implique une taille identique.

Par exemple, dans le cas le plus général, on va chercher à faire correspondre la matrice de taille 3x2 (colonnes x lignes) d'une lettre supposée 'L' avec la matrice de taille 2x3 de la lettre 'L' que l'on a préalablement apprise.

Le principe du recadrage consiste à rechercher le plus petit commun diviseur (PPCD) du nombre de lignes et celui du nombre de colonnes des deux matrices :

Matrice originale : 3x2

PPCD(3,2)=6 (lignes)

PPCD(2,3)=6 (colonnes)

Taille de la matrice intermédiaire : 6x6

Matrice finale : 2x3

Matrice originale : 3x2

1		
1	1	1

Matrice intermédiaire : 6x6

1	1				
1	1				
1	1				
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

Matrice finale : 2x3

2/3	
5/6	1/2
1	1

Lorsque la taille de la matrice de la lettre candidate est grande et celle finale petite, on s'écarte peu du recadrage exact en divisant simplement la taille de la première par celle de la seconde, successivement pour les lignes et les colonnes. En pratique, cela permet de réduire sensiblement le temps de calcul.

On remarque que dans la matrice finale, l'ajustement matriciel engendre des valeurs réelles et non plus entières dans chaque cellule : ces valeurs peuvent être assimilées à un niveau de remplissage des "pixels", c'est-à-dire que l'on passe de pixels binaires à des pixels à niveaux multiples (niveau de gris par exemple).

Ce phénomène est dû à la modification de l'échantillonnage de l'image. A ce stade, l'objectif est de réduire le plus possible la perte d'information avant l'étape de binarisation (seuillage binaire) : on cherche donc à ce que le contraste soit le plus élevé possible, les images binaires étant par définition les plus contrastées ; cependant, la taille de la matrice finale est définie à l'avance ; le contraste n'est par conséquent pas modifiable. Le seul moyen de parvenir à un bon contraste est de prédéfinir plusieurs tailles de matrices de référence et de retenir celle qui donnera le meilleur contraste.

Par exemple, si l'on dispose de quatre tailles de matrice différentes : de 3x4 à 4x5, on va d'abord rechercher quel est le nombre de colonnes offrant le plus grand contraste, puis de la

même façon le nombre de lignes ; la matrice optimale obtenue pour la lettre 'B' est 3x5 (ou 4x5), en revanche les matrices 3x4 et 4x4 sont inadaptées :

1	1	
1		1
1	1	
1		1
1	1	

Nous présentons maintenant un exemple d'apprentissage à partir de ces matrices.

2. Apprentissage

2.1. Principe de l'apprentissage

Considérons l'apprentissage de la lettre 'A' sur une matrice 3x5 constituée des pixels binaires de a à p :

a	b	c
d	e	f
g	h	i
k	l	m
n	o	p

On suppose que l'on dispose de plusieurs représentants similaires mais pas tout à fait identiques de la lettre 'A' ; le nombre minimum de représentant est 1 et le nombre maximum ainsi que la variation maximale sont fonction de la discrimination qui en résulte sur la totalité de la base d'apprentissage (de 'A' à 'Z') ; ce cas idéal dépend de la quantité d'information contenue dans la matrice, c'est-à-dire du nombre de bits codant les pixels.

Les différents modèles du 'A' devront être les plus différents possible (pour couvrir la plus grande gamme de 'A'), mais chacun d'eux devra toujours être plus proche de la classe des 'A' plutôt que de tout autre classe de lettres.

Pour chaque pixel d'un caractère on étudie les deux cas suivants :

- le pixel a peut il être présent dans la lettre 'A' (dans la classe des 'A' appris) ?

1 : Oui, 0 : Jamais présent dans aucun des représentants des 'A'

- le pixel a peut il être absent dans la lettre 'A' ?

1 : Oui, 0 : Jamais absent dans aucun des représentants des 'A'

- le pixel b peut il être présent dans la lettre 'A' ?

1 : Oui, 0 : Jamais présent

- le pixel b peut il être absent dans la lettre 'A' ?

1 : Oui, 0 : Jamais absent

...

- le pixel a peut il être présent dans la lettre 'B' ?

1 : Oui, 0 : Jamais présent

- le pixel a peut il être absent dans la lettre 'B' ?

1 : Oui, 0 : Jamais absent

...

On utilise donc 2 bits (1 pour les présences et 1 pour les absences) pour chaque cellule de la matrice et pour chaque lettre, soit pour un ensemble de référence codant les 8 premières lettres de l'alphabet (de 'A' à 'H'), 2 octets x 3 x 5 = 30 octets pour des matrices 3x5.

Pour coder les 26 lettres de l'alphabet (avec 6 caractères libres, 26 + 6 = 32 soit 4 blocs de 8 lettres) il suffira donc de 30 x 4 = 120 octets.

2.2. Encodage de l'apprentissage

Pour passer d'une matrice floue (à niveaux multiples) obtenue par le recadrage matriciel à une matrice de pixels à niveau binaire (0 ou 1), on effectue une opération de seuillage, une binarisation.

Après la première perte d'information lors du recadrage (due à la modification de l'échantillonnage sur la taille des matrices), on perd une seconde fois de l'information par cette opération de seuillage (qui est due à la modification de l'échantillonnage cette fois sur les niveaux des pixels) ; afin de minimiser cette perte, on va tenir compte de la densité moyenne des niveaux des pixels de la matrice : si le caractère est tracé en style gras, alors cette densité sera élevée, tandis que si les caractères sont de type bâton en tracé fin, la densité sera faible. Un seuil relatif à cette densité sera donc stable dans tous les cas d'épaisseur et de taille de caractère.

De plus, on effectue deux fois l'opération de seuillage avec deux seuils distincts (un seuil bas et un seuil haut) et on considère les deux caractères binaires obtenus comme s'il s'agissait de deux variantes du même caractère. Cela permet par exemple d'extraire deux représentants de la lettre 'A' sur des matrices 3x4 à partir de la même image en 6x8 :

	1	1	1	1	
1	1	1	1	1	1
1	1			1	1
1	1			1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1			1	1
1	1			1	1

	1	
1		1
1	1	1
1		1

1	1	1
1		1
1	1	1
1		1

3. Reconnaissance

Une fois l'apprentissage accompli sur l'ensemble des caractères de la base de référence, pour chaque taille de matrice, la reconnaissance est alors opérationnelle :

Pour chaque caractère candidat à la reconnaissance, on effectue l'ajustement matriciel correspondant au contraste maximum (cf. § 1.). Sur la matrice obtenue par ajustement matriciel, on procède à un seuillage à partir d'un seuil moyen relatif à la densité moyenne (le seuil moyen est compris entre le seuil bas et le seuil haut). Enfin, il suffit de faire une comparaison point à point ("matching") des pixels de cette matrice binaire avec tous les caractères de la base, pour cette même taille de matrice. Cette procédure, classique en RdF, est opérée par un ET-logique entre les bits correspondants de chaque matrice. Les bits codant les pixels présents de la base de référence seront utilisés si leur correspondant code un pixel présent de la matrice du caractère candidat, sinon, ce seront les bits codant les pixels absents qui seront utilisés pour le masque.

La programmation de cette procédure de masquage s'effectue de la manière suivante :

Une variable résultat, codée sur un seul octet, est initialisée à la valeur 255 (tous les bits actifs) ; si la variable résultat vaut zéro à la suite des 15 ET-logiques correspondant aux 15 pixels d'un caractère de taille 3x5, alors il ne peut s'agir d'aucune lettre appartenant au bloc des 8 lettres 'ABCDEFGH' ; sinon, il faut décomposer la variable résultat en puissance de deux pour retrouver quelle(s) lettre(s) correspond(ent) au caractère candidat.

On procède de la même façon pour les autres blocs de 8 lettres 'IJKLMNOP', 'QRSTUVWXYZ' et 'YZ-----'.

A l'issue de cette phase, plusieurs cas se présentent :

1^{er} cas : un seul caractère de référence correspond exactement au caractère candidat : celui-ci est donc "reconnu".

2^{ème} cas : aucun caractère de référence ne correspond exactement au caractère candidat : par exemple, l'apprentissage n'a été réalisé que sur un seul ensemble de caractère de référence et celui-ci ne correspond pas tout à fait avec celui à reconnaître présentement.

3^{ème} cas : plusieurs caractères de référence correspondent exactement au caractère candidat : dans ce cas, il est clair que la capacité d'apprentissage de la base a été dépassée (saturation), au moins pour la taille de matrice employée, puisqu'une même configuration de pixels correspond à plusieurs caractères. Ce manque de discrimination trahit la trop grande quantité d'information que l'on a proposée par rapport à ce que peuvent enregistrer les matrices d'apprentissage. Il faut donc utiliser des matrices de taille plus grande (cf. § 5. Tests d'apprentissage).

Quel que soit le cas de figure qui se présente, aucune décision ne doit être prise hâtivement : la procédure de décision que nous présentons va définir dans quel cas on peut conclure à la reconnaissance d'un caractère en prenant le minimum de risque d'erreur, même si l'apprentissage est en saturation (cas n°3).

4. Décision

4.1. Détermination d'un tableau de score

Au lieu d'un simple masquage de matrice, on calcule une fonction score définie par la somme des pixels coïncidant avec les pixels correspondant du caractère candidat. Cette fonction prend son utilité surtout dans les cas 1 et 2, car dans le 3^{ème} cas, on ne pourra de toute façon pas trancher entre tous les caractères retenus par le masquage, et seule une analyse approfondie le permettra (cf. § 4.3.), c'est-à-dire par une accumulation d'indices de reconnaissance obtenus avec des matrices de tailles différentes.

On obtient alors une suite de "scores" correspondant à chaque lettre de l'ensemble de référence de 'A' à 'Z'. Le score maximum est obtenu par toutes les lettres déjà sélectionnées à l'issue du masquage, et le score nul, qui est un cas assez rare, correspond aux lettres dont l'intersection matricielle avec le candidat est nulle.

À ce stade, l'idée la plus simple consisterait à décider la reconnaissance de la lettre qui a le score le plus élevé. Cependant, en RdF, le taux de reconnaissance n'est intéressant que si le taux de rejet est faible : ce dernier est défini par le taux d'erreurs de reconnaissance et correspond aux caractères mal identifiés pour lesquels il aurait mieux valu conserver une indécision (caractère '?').

Ainsi, il est plus judicieux de définir d'abord le taux de rejet maximum que l'on peut tolérer, et ensuite de reconnaître le maximum de caractères possible sous cette contrainte.

Tout le problème consiste donc à déterminer une fonction de décision dont le but est de décider que le caractère de score le plus élevé est bien celui à reconnaître en commettant une erreur de décision inférieure ou égale à $e\%$ (par exemple, si $e=10\%$, la fonction ne doit pas se tromper plus d'une fois sur 10).

Cette seule procédure suffit dans la plupart des cas où les primitives extraites discriminent assez bien les caractères, mais lorsque les caractères sont ambigus, il peut être intéressant d'extraire plus d'information sur le tableau de score plutôt que de décider systématiquement pour la reconnaissance d'un seul caractère ou bien l'échec (la fonction indique alors '?'). En commettant toujours le minimum d'erreurs de décision, on va rechercher l'ensemble des caractères probables.

Dans un texte, la succession des caractères n'est pas quelconque mais restreinte à un dictionnaire, de sorte qu'une certaine ambiguïté pourra être tolérée dans la reconnaissance des caractères dans la mesure où cette ambiguïté pourra facilement être levée à un niveau plus élevé du processus de la reconnaissance. Dans le cas où la succession des caractères est quelconque, ou dans le cas où un dictionnaire adapté n'existe pas, il est bien sûr inutile et inexploitable de savoir quelles lettres sont plus ou moins voisines.

En outre, lorsque les primitives ne sont pas assez discriminantes, si l'on utilise plusieurs algorithmes spécialisés dans l'étude d'une seule caractéristique particulière, il est utile de pouvoir accumuler tous les indices ou évidences concernant un ensemble de caractères tous probables au moins à $e\%$ près.

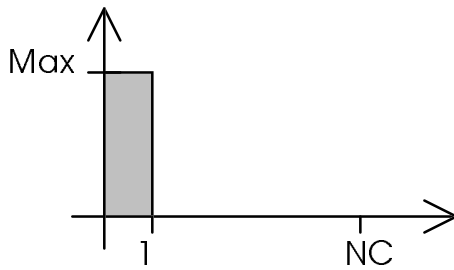
4.2. Définition d'une fonction de décision

Nous allons établir cette fonction à partir du tableau de score, par l'étude de l'allure qu'elle doit respecter dans les cas de figures extrêmes.

Max : Score Maximum du tableau, $\text{max} = \text{nombre de lignes} * \text{nombre de colonnes de la matrice}$;

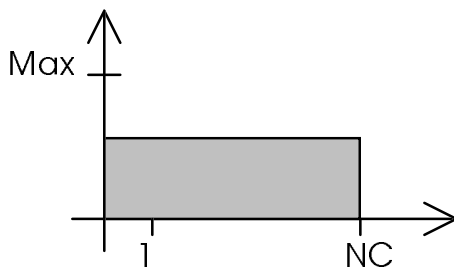
NC : Nombre de caractères de la base de référence ;

Cas de figure n°1 : Décision certaine



Dans ce cas n°1, on est sûr à 100% (erreur 0%) que le caractère est identifié (caractère de score le plus élevé : n°1).

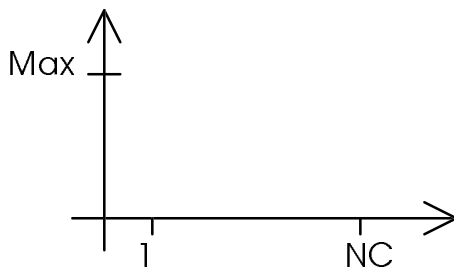
Cas de figure n°2 : Indécision absolue



On ne dispose d'aucun indice, tous les caractères de la base de référence sont équiprobables : toute décision sera totalement hasardeuse.

Cas de figure n°3 :

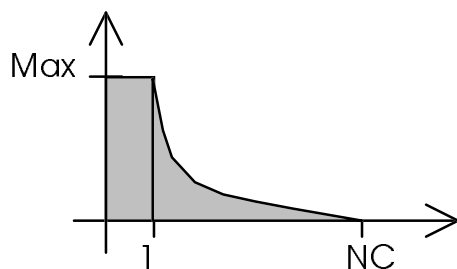
Base d'apprentissage vide



On est sûr que le caractère ne correspond à rien du tout : il est nouveau à 100%.

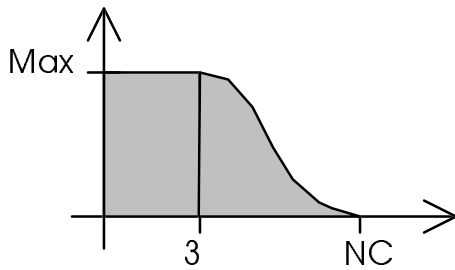
Dans la réalité, on se trouve en présence des cas suivants :

Cas de figure n°4 : cas normal



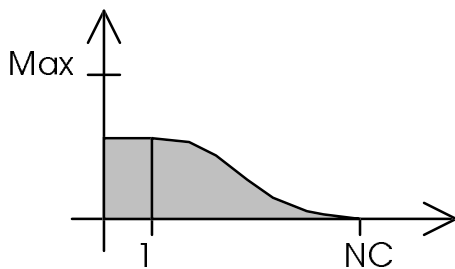
C'est le cas normal, dans lequel le caractère n°1 est probablement reconnu à une erreur de e% près.

Cas de figure n°5 : Ex aequo



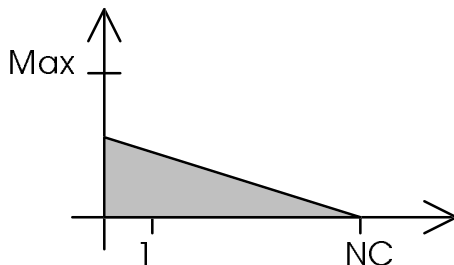
1^{ère} ambiguïté : Trois caractères sont probables (à $\epsilon\%$ près) sans qu'il soit possible d'en choisir 1 plutôt que les deux autres (la méthode de l'accumulation d'indices est applicable).

Cas de figure n°6 : Indécision



2^{ème} ambiguïté : Plusieurs caractères se rapprochent du caractère candidat, mais sans plus (accumulation d'indices applicable).

Cas de figure n°7 : Forte indécision



3^{ème} ambiguïté : Aucun caractère ne se dégage du lot, il y a seulement un classement incertain.

On définit une plausibilité de reconnaissance qui est une estimation de la probabilité de reconnaître un caractère. Elle tient compte non seulement du meilleur score d'appariement, c'est-à-dire le coût le plus faible, mais également de l'écart relatif entre les premiers scores.

Si un écart important distingue nettement le premier, ou le groupe des deux ou trois premiers caractères, du reste de l'alphabet dans le classement, alors on considère que la plausibilité de reconnaissance est élevée.

En revanche, si aucun caractère ne se dégage du lot par un score plus élevé, on considère que la plausibilité est faible.

Dans la suite, d représente le nombre maximum de caractères ambigus que l'on peut sélectionner.

4.3. Décision de reconnaissance

On définit les deux seuils de décision suivant :

- le premier seuil est le nombre maximum de caractères pour lequel la valeur d conserve un sens (par exemple on considère qu'au delà de 20% du nombre de caractères de la base, l'information représentée par la décision est vraiment inutilisable) ;

- le second seuil est la plausibilité minimale exigée pour que le caractère à reconnaître se trouve parmi le groupe des d premiers sélectionnés.

La première décision qui est prise est la suivante :

```
Si  $d$  vaut 1 et que la plausibilité  $P[1]$  est supérieure à la
plausibilité minimale, le processus est terminé et le caractère est
reconnu ;
Sinon, lorsque  $d > 1$ , on poursuit le processus quelle que soit la
valeur de  $P[d]$ , en examinant les matrices de toutes tailles qui
offrent un contraste suffisant (contraste supérieur au seuil
MIN_CONTRASTE).
```

En appliquant la fonction de décision dans chaque cas, on calcul le score de chaque caractère en cumulant les scores sur toutes les matrices en pondérant par le contraste et la plausibilité $P[d]$ globale du groupe normalisée par la dimension de la matrice (division par le nombre de pixels de la matrice correspondante).

Enfin, en transformant la fonction score en une fonction coût et en la classant par ordre croissant, on applique une nouvelle fois la même fonction de décision :

```
Si  $d$  vaut 1 et que la plausibilité  $P[1]$  est supérieure à la
plausibilité minimale, le processus est terminé et le caractère est
reconnu ;
Sinon, si la plausibilité est inférieur ou bien si  $d$  est supérieur
au nombre maximum de caractères significatifs, la fonction de
décision conclut à un échec de reconnaissance (le caractère '?' est
retourné)
Sinon, on sélectionne la liste des  $d$  premiers caractères.
```

Toutefois, le mode de décision consistant à ne répondre que le caractère de score le plus élevé à également été prévu et dans ce cas, la décision est la suivante :

Si la plausibilité de reconnaissance $P[1]$ du caractère de score le plus élevé est supérieure à la plausibilité minimale, on retourne ce caractère ;
Sinon, la fonction de décision conclut à un échec de reconnaissance (caractère '?').

5. Tests d'apprentissage et de reconnaissance

5.1. Tests d'apprentissage

Nous avons testé la capacité d'apprentissage du système sur un ensemble de caractères bâton utilisés par l'INSA pour la saisie automatique des dossiers d'inscription des étudiants candidats à l'entrée à l'école d'ingénieurs (cf. [Chapitre III § 3.3.](#)). Pour répondre à une nécessité de rigueur en reconnaissance de caractères manuscrits omniscriteurs, une forte contrainte sur l'écriture a été imposée aux étudiants : l'ensemble des caractères utilisables est restreint à l'ensemble suivant :

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9 9 6
/ - / /

Pour l'apprentissage, les chiffres 6 et 9 ont été doublés du fait du grand nombre d'erreurs commises par les candidats.

La saisie est effectuée au moyen d'un scanner de résolution de 200 dpi soit 54 pixels de haut sur 54 pixels de large par caractère.

En utilisant une seule taille de matrice pour l'apprentissage et la reconnaissance, le système est capable de discriminer tous les caractères quelle que soit la taille des matrices comprise entre 4 et 8 colonnes et entre 5 et 8 lignes. L'apprentissage minimal est donc assuré à partir d'une taille de matrice de 4x5 (colonnes x lignes).

En deçà de cette taille, les premières ambiguïtés non solubles surviennent pour les lettres 'N' et 'M' sur les tailles 3x5, ainsi que pour les caractères 'G' et '6', '0QBO', 'Z' et 'I'.

Lorsque l'on descend à 2 colonnes sur 6 lignes, seuls les caractères 'FJLZ1235679' sont discriminés. Si on réduit le nombre de lignes à 3 et le nombre de colonnes à 6, des ambiguïtés se produisent, notamment sur les caractères 'FP', 'RE', 'YV', 'ZIS', '96E'...

5.2. Tests de reconnaissance

Nous avons repris une quinzaine de dossiers de candidature fictifs qui ont déjà fait l'objet de tests avec le système actuel de reconnaissance de l'INSA. Les images obtenues après la scannérisation sont de la forme suivante :

```

DARGENTON, PATRICE
CHATILLON/THOUET

79200 PARTHENAY

49951899
M          080168          F
          X
X          4          3          2
```

Sur les 683 caractères lus, 645 ont été reconnus, soit un taux de 94%.

Parmi les erreurs observées, on remarque que certains caractères sont contraignants pour les scripteurs, notamment le chiffre '1' qui a souvent été tracé sans la barre en dessous. L'ajout de cette variante du '1' dans la base d'apprentissage a permis de résoudre ce défaut.

On relève une part d'erreur due aux ratures apparaissant surtout sur les caractères contraignants tels que 'D', '8' et 'S'.

D'autre part, la lettre 'R' est très proche du 'A' et ces lettres sont souvent confondues, de même pour les couples 'G' et '6', 'Q', '0' et 'D', 'B' et '8', '5' et 'S'.

6. Bilan et conclusion sur la reconnaissance matricielle des caractères

6.1. Domaine d'application

Dans la mesure où le processus de la segmentation de l'image en caractère est accomplie correctement (cf. Chapitre II. Segmentation des Documents), la RMC peut s'appliquer aux caractères imprimés simples (majuscules d'imprimerie, caractères OCRA ou OCRB utilisés couramment dans l'industrie) et bruités (cf. § 6.2. Tolérances), ainsi qu'aux

caractères manuscrits omniscriteurs sous forte contrainte (caractères bâton tracés dans des cases...). Cependant, la discrimination entre les caractères est insuffisante pour traiter la variété des documents imprimés (cf. § 5.2. Tests de reconnaissance), et seuls des logiciels spécifiques peuvent les traiter en procédant dans la plupart des cas par apprentissage des fontes employées sur chaque document.

6.2. Tolérances

La RMC est insensible à la taille et à l'épaisseur du tracé des caractères grâce au recadrage matriciel (cf. § 1. Ajustement matriciel) ; elle ne peut donc pas différencier les lettres 'e' de 'l', 'u' de 'U' ni ' ' de '/'. Cependant, en calculant le ratio entre la hauteur du cadre englobant et la hauteur moyenne des caractères lus, ainsi que la position moyenne de l'ordonnée du début des caractères par rapport à la ligne de texte entière, une comparaison avec ces mêmes valeurs calculées au moment de l'apprentissage permet de pallier simplement ce défaut.

La méthode tolère le bruit tant que celui-ci n'empêche pas la discrimination entre les caractères de référence. Le bruit peut être diffus, en petites taches, ou bien provenir de l'imprimante : par exemple, les références des produits de consommation imprimés sur l'emballage sont souvent dégradées du fait de la difficulté d'impression. Les caractères peuvent être formés de plusieurs connexités (tracé par "points ronds") ce qui rend leur segmentation plus difficile, cependant, la reconnaissance ne sera pas affectée.

D'une façon générale, le bruit sera en grande partie éliminé par le recadrage et la binarisation, alors que celui-ci peut affecter d'autres méthodes de reconnaissance, notamment la reconnaissance structurelle.

6.3. Sensibilités et intolérances

Une variation de l'épaisseur du tracé sur un même caractère empêcherait une binarisation correcte de la matrice recadrée, car il n'y a pas de seuillage adaptatif local. Cependant, ce cas de figure est assez marginal et il existe des défauts plus importants.

Toute transformation de type cisaillement (même caractère en italique) ou bien de type déformation non linéaire empêche la mise en correspondance rectangulaire des caractères. Par exemple, une disproportion d'une boucle du chiffre '6' ou '9' par rapport au cadre englobant le caractère entraînera ce problème, tandis qu'une méthode de reconnaissance de type structurel est tout à fait indiquée pour traiter ce cas.

En fait, il suffirait de modéliser la transformation non linéaire et de l'appliquer sur la grille matricielle plaquée sur le caractère ("Elastic matching") pour que tout le processus reste valide mais on perdrait alors la simplicité de mise en oeuvre.

6.4. Capacité

La RMC offre une certaine capacité de généralisation inhérente à la procédure d'apprentissage que l'on peut observer dans l'exemple suivant :

En apprenant deux représentants de la lettre 'C' : 'C' et 'C', on peut reconnaître une troisième version de cette lettre 'C' qui n'a pas été apprise, mais qui est une combinaison des caractéristiques des deux premières.

Présence

1	1	1
1	0	1
1	0	0
1	0	1
1	1	1

Absence

1	0	1
0	1	1
0	1	1
0	1	1
1	0	1

Cependant, cette généralisation est limitée par la quantité d'information contenue dans les matrices d'apprentissage, mais aussi par la nécessité de discriminer les caractères de l'ensemble de référence.

On peut reprocher à cette méthode que l'apprentissage permet de mémoriser seulement des pixels indépendants les uns des autres : les deux pixels représentés en style gras sur la matrice d'absence ci-dessus ne doivent pas être absents simultanément par rapport aux deux lettres 'C' apprises alors que cela n'est pas effectivement contraint par l'apprentissage. En théorie, on pourrait considérer l'ensemble des combinaisons des pixels, mais cet ensemble croît si vite avec la taille des matrices que leur mémorisation s'avère prohibitive :

Matrice de taille $3 \times 4 = 12$ pixels $\rightarrow 2^{12} = 4096$ configurations distinctes de pixels binaires 0/1.

Matrice de taille $3 \times 5 = 15$ pixels $\rightarrow 2^{15} = 32768$ configurations distinctes de pixels binaires 0/1.

Matrice de taille $4 \times 6 = 24$ pixels $\rightarrow 2^{24} = 16$ millions de configurations distinctes de pixels binaires 0/1.

En pratique, on ne choisira que quelques primitives pertinentes par rapport au cas particulier de reconnaissance telles que les lignes droites, arcs, boucles, concavités, les primitives topologiques, gradients, histogrammes simples ou de transition... (cf. [Chapitre I](#)).

6.5. Conclusion

L'extraction de primitives a donc pour objet de réduire la quantité d'information par le changement de représentation en espérant que la plus grande partie du bruit ainsi que les éventuelles variations entre le caractère à reconnaître et celui appris, seront absorbées.

Pour la RMC, l'extraction de primitives est réduite à sa plus simple expression, ce qui en fait un algorithme rudimentaire parmi les nombreux algorithmes de reconnaissance de caractères existant.

En revanche, l'étape de décision peut être commune à tous et il y a un intérêt à mettre en collaboration autant d'algorithmes différents et complémentaires que possible. La difficulté consiste à pondérer les résultats de chacun en fonction de la pertinence et de la discrimination observée dans une application particulière.

Il suffit alors de prendre la décision finale qui peut faire ressortir le caractère à reconnaître grâce à l'accumulation d'indices.

Table des matières

<i>Introduction</i>	1
<i>1. Ajustement matriciel</i>	1
<i>2. Apprentissage</i>	3
2.1. Principe de l'apprentissage.....	3
2.2. Encodage de l'apprentissage.....	4
<i>3. Reconnaissance</i>	5
<i>4. Décision</i>	6
4.1. Détermination d'un tableau de score.....	6
4.2. Définition d'une fonction de décision.....	7
4.3. Décision de reconnaissance.....	10
<i>5. Tests d'apprentissage et de reconnaissance</i>	11
5.1. Tests d'apprentissage.....	11
5.2. Tests de reconnaissance.....	12
<i>6. Bilan et conclusion sur la reconnaissance matricielle des caractères</i>	12
6.1. Domaine d'application.....	12
6.2. Tolérances.....	13
6.3. Sensibilités et intolérances.....	13
6.4. Capacité.....	14
6.5. Conclusion.....	15
<i>Table des matières</i>	15