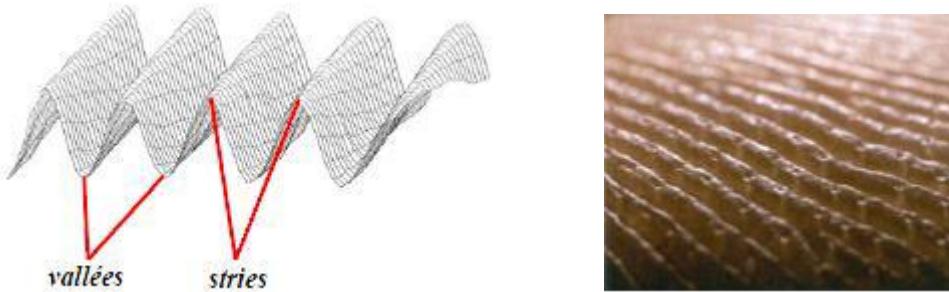


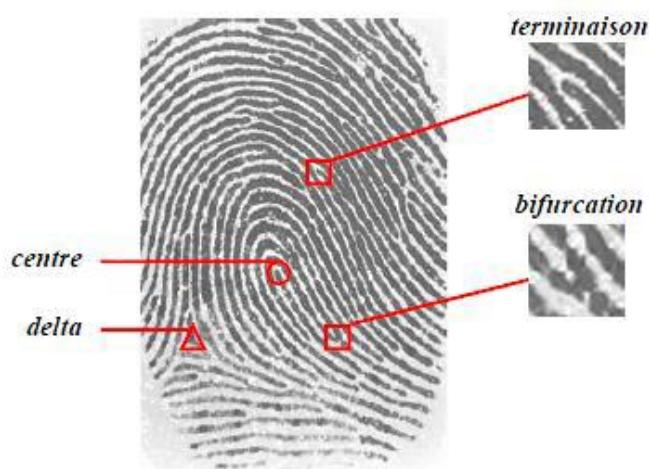
Chapitre III : La biométrie d'empreinte digitale : identification et authentification des individus

1. Définition

Une empreinte digitale se compose de beaucoup des rides appelées **crêtes** (stries) et d'espace entre les crêtes appelés **vallées**. Les caractéristiques d'empreintes digitales sont les minuties qui sont les points anormaux sur les rides.



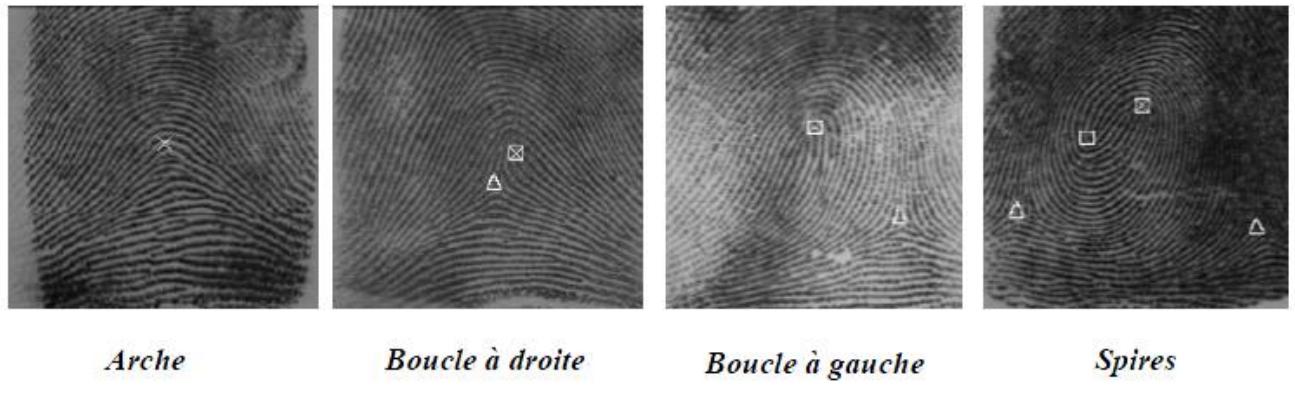
- Chaque empreinte possède un ensemble des points singuliers globaux (*les centres ou noyaux et les deltas*) et locaux (*les minuties*) [1]. Les centres correspondent à des lieux de convergences des stries tandis que les deltas correspondent à des lieux de divergences.
- 2 types de minuties sont les plus utilisés, ce sont *l'arrêt de ride (terminaison)* et *la bifurcation* [2][6].



- La position et le nombre de centres et de deltas permettent de classifier les empreintes en catégorie. On distingue principalement 3 grandes familles [5] selon *Gelton* :

- * **Arches** ou tentes (regroupe 5% des doigts humains)
- * **Boucles à droite** ou **boucles à gauche** (regroupe 65% des doigts humains)
- * **Spires** ou tourbillons (regroupe 30% des doigts humains)

Il y a deux spirales qui sont des doubles boucles imbriquées, et elles sont assez rares.



- Pour rendre la reconnaissance d'empreinte très efficace, on procède à la classification de notre base de donnée en sous classes selon les types des empreintes existants.

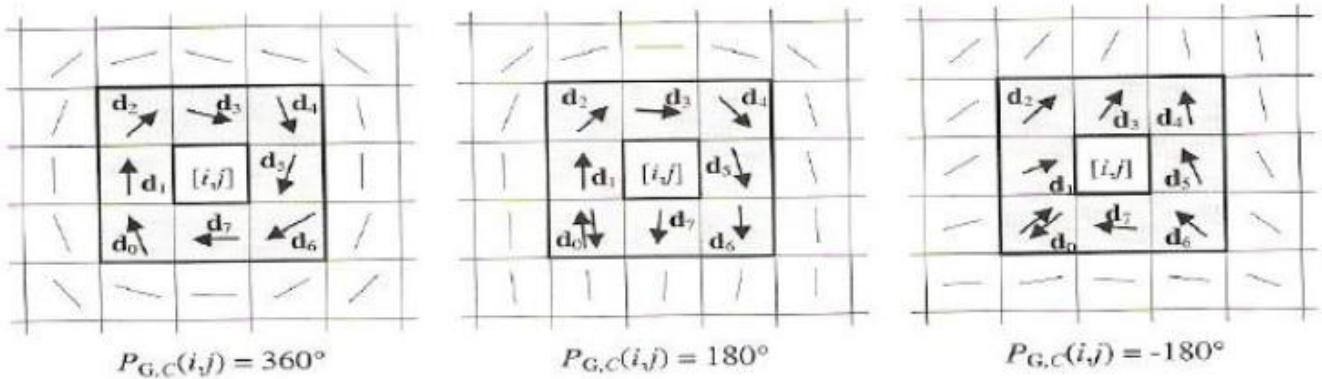
Les stries adoptent certaines formes distinctes. Ces formes s'appellent des régions caractéristiques ou régions singulières ; elles sont classées en trois grandes catégories : **boucle**, **delta** et **spires** ou tourbillon. Les régions singulières sont utilisées dans la classification des empreintes digitales.



- L'idée de la classification consiste à détecter les caractéristiques globales (les régions singulières) d'une empreinte digitale, l'un des algorithmes les plus pratiques c'est l'algorithme de *l'index de Poincaré* proposé par Kawagoe et Tojo. Son principe est le suivant :

Soit un pixel $[i,j]$ de la matrice D; l'index de Poincaré $P_{G,C}(i,j)$ est calculé par la formule suivante :

$$P_{G,C}(i,j) = \sum_{k=0}^7 \text{angle}(\mathbf{d}_k, \mathbf{d}_{(k+1)\bmod 8})$$



Alors :

1. Si $P_{G,C} = 0^\circ$, $[i,j]$ ne se trouve pas dans une région singulière
2. Si $P_{G,C} = 360^\circ$, $[i,j]$ se trouve dans une région singulière en spire (centre)
3. Si $P_{G,C} = 180^\circ$, $[i,j]$ se trouve dans une région singulière en boucle
4. Si $P_{G,C} = -180^\circ$, $[i,j]$ se trouve dans une région singulière delta

La classification Galton-Henry stipule que :

- Une empreinte $\in \{\text{la classe arche}\}$ si elle ne contient ni boucle ni delta comme régions singulières ;
- Une empreinte $\in \{\text{la classe boucle}\}$ si elle contient une boucle et un delta, les boucles qui entrent et ressortent du côté gauche sont dites des boucles à gauche tandis que celles qui rentrent et ressortent du côté droit sont dites des boucles à droite ;
- Une empreinte $\in \{\text{la classe spires}\}$ si elle contient deux régions singulières boucles et deux régions deltas.

2. Conception du système de reconnaissance des empreintes digitales

Un système de reconnaissance des empreintes digitales est un système automatique de reconnaissance de formes qui se compose des étapes suivantes :

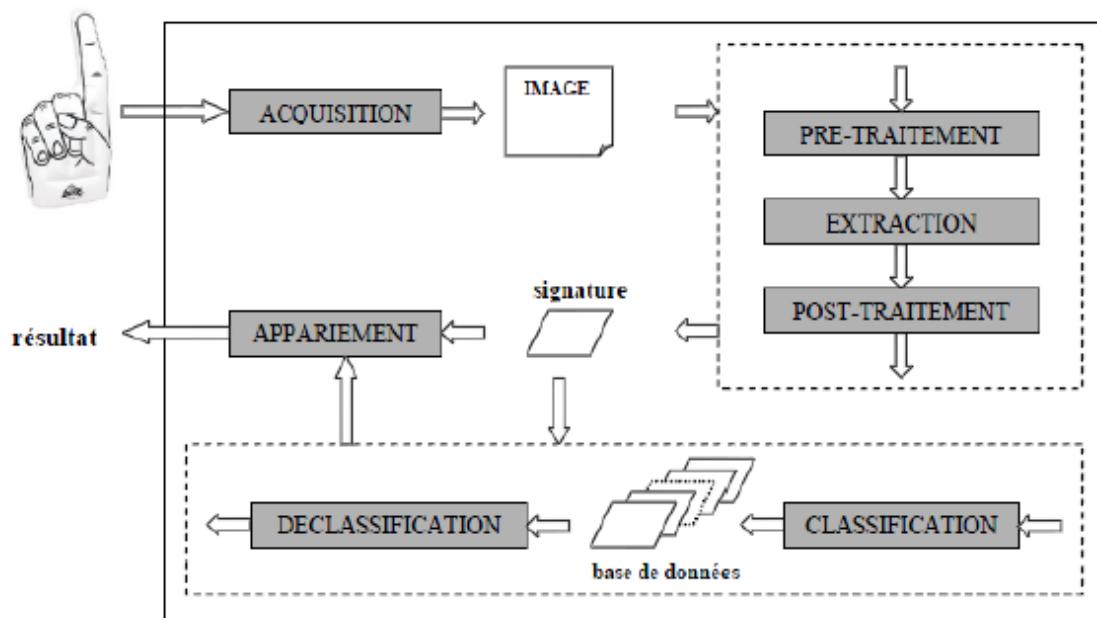


Figure : Architecture générale d'un système complet de reconnaissance d'empreintes

2.1 L'acquisition de l'image d'empreinte

Il existe 2 méthodes pour l'acquisition d'une image d'empreinte digitale :

→ **L'acquisition indirecte** : où il existe 2 méthodes :

* **L'empreinte acquise par encre** :

Après l'avoir enduit l'encre, le doigt est imprimé sur un bout de papier. Ce papier passe ensuite au scanner standard pour être numérisé.



* **les empreintes latentes** :

Elles sont formées suite à une légère trace laissée sur un objet due à la sécrétion constante de la sueur. Elles sont prises à l'aide d'une poudre spéciale.



→ **L'acquisition directe** : ce sont les capteurs, où on peut citer :

Les capteurs optiques (les plus répandus, dont le principe de fonctionnement est le même que celui du scanner ainsi ses prix acceptables), les capteurs thermiques, les capteurs à ultrasons qui sont assez chers, les capteurs intégrés au silicium.



2.2 Pré-traitement

Pour améliorer la qualité des images d'empreintes digitales, on fait un pré-traitement qui se compose de lissage, d'amélioration de contraste, de filtrage.

2.3 Extraction des caractéristiques

L'ensemble des minuties est extrait à partir de l'image filtrée. Pour cela nous étudierons 2 méthodes :

- La méthode classique consiste à extraire les minuties sur un squelette binaire de l'image filtrée
- La méthode directe consiste à extraire les minuties directement sur l'image filtrée

2.3.1 La méthode directe

Peu d'approches d'extraction de minuties suivant ce schéma ont été proposées :

- *M.T.Leung et al* introduisent un réseau de neurones pour la détection des minuties.
- Une autre approche plus directe a été proposée par *Maio et Maltoni* [3]. Cette approche traque les crêtes en localisant les maxima locaux par un suivi de lignes le long du flot directionnel des crêtes pour détecter les minuties :

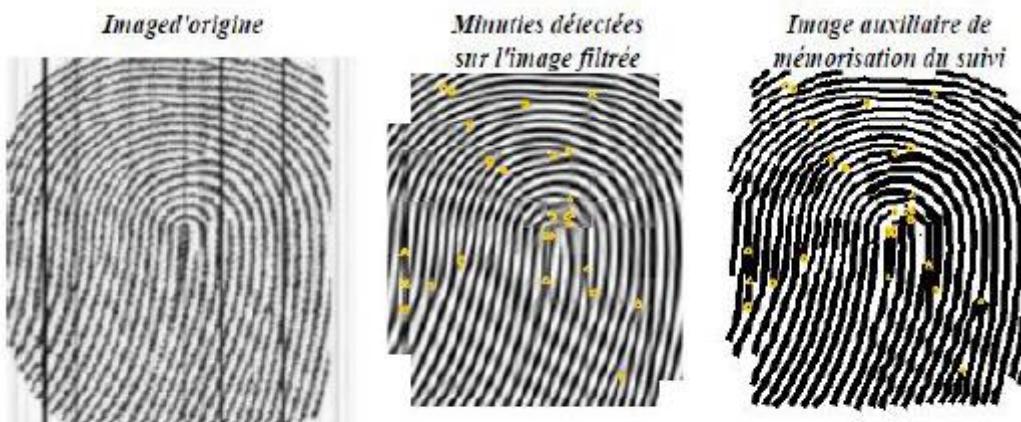


Figure 1 : Détection des minuties par le suivi des stries de l'image filtrée

2.3.2 La méthode classique

Le processus classique entier d'un algorithme d'extraction de minuties est montré dans la figure suivante [2][6], il est proposé par *Jain et Al* :

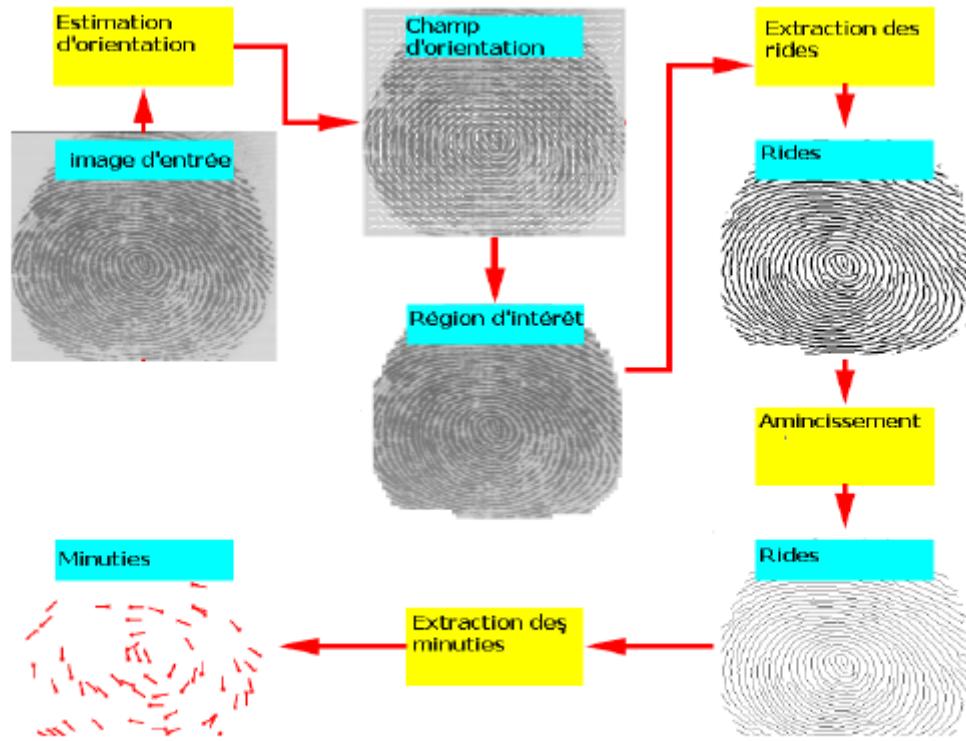


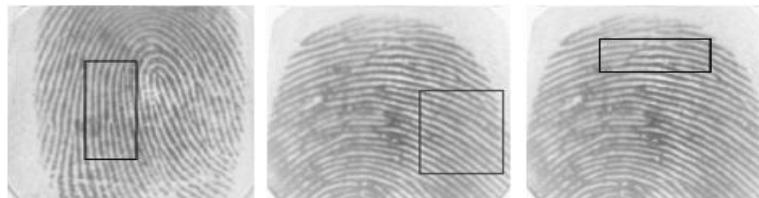
Figure : Organigramme de l'algorithme d'extraction des minuties proposé par Jain et al

a) Estimation de champ d'orientation :

Elle consiste à avoir une estimation de l'orientation des lignes d'une empreinte digitale sur une petite zone de l'image. Cette estimation est reprise dans une matrice notée D , appelée **image directionnelle** ou **orientation de l'empreinte digitale**.

La case (i,j) de D contient la valeur de l'angle $\theta_{i,j}$ entre l'axe horizontal et l'orientation des lignes de l'empreinte dans le voisinage du pixel $[i,j]$.

La manière la plus simple à calculer D consiste à calculer le gradient aux différents points de l'image originale E de l'empreinte.



Parmi les algorithmes pour faire ce calcul, y a la méthode de **Anil Jain et al** qui consiste à diviser l'image en plusieurs fenêtres de taille $W \times W$. Pour tous pixels dans chaque fenêtre, on calcule les gradients G_x et G_y , puis on calcule l'orientation locale de ce pixel par les formules [2]:

$$V_x(i,j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} (2G_x(u,v)G_y(u,v))$$

$$V_y(i,j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} (G_x^2(u,v) - G_y^2(u,v))$$

$$\theta(i,j) = \frac{1}{2} \tan^{-1} \left(\frac{V_x(i,j)}{V_y(i,j)} \right)$$

Y a aussi l'algorithme de **Sobel** qui est très utilisé. Son principe est le suivant : Il calcule le gradient de l'intensité en chaque pixel par rapport à x et y comme suit :

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * E \quad \text{et} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * E$$

Puis l'angle du gradient qui nous donne l'orientation de lignes de l'empreinte au voisinage d'un pixel $[x_i, y_j]$ est donné par :

$$\theta_{i,j} = \arctan (G_x[x_i, y_j] / G_y[x_i, y_j])$$

b) Segmentation :

Après avoir estimé le champ d'orientation de l'image d'empreinte digitale, un algorithme de segmentation qui est basé sur le niveau de certitude du champ d'orientation est employé pour localiser la région d'intérêt dans l'image.

Le niveau de certitude du champ d'orientation au pixel (i,j) est défini comme suit [2] :

$$CL(i,j) = \sqrt{\frac{1}{W^2} \frac{V_x^2(i,j) + V_y^2(i,j)}{V_e(i,j)}}$$

$$\text{Tel que : } V_e(i,j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} (G_x^2(u,v) + G_y^2(u,v))$$

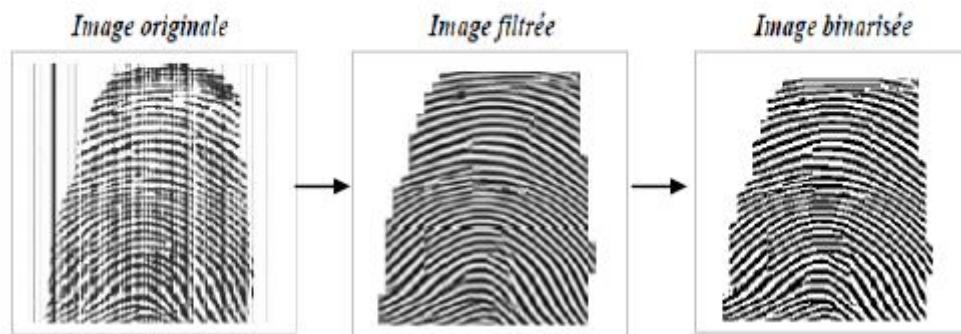
Si : $CL(i,j) < \text{le seuil } T \implies \text{le pixel est marqué comme pixel de fond}$

Si : $CL(i,j) > \text{le seuil } T \implies \text{le pixel est marqué comme un pixel de la région d'intérêt}$

c) La détection de rides (binarisation) [3]:

Consiste à transformer l'image de 256 niveaux de gris en image binaire où les pixels noirs correspondent aux crêtes et les pixels blancs aux vallées. La technique la plus utilisée consiste à se fixer un seuil d'intensité T , puis parcourir la matrice de pixels de l'image en se basant sur la procédure suivante :

$$I_T(x,y) = \begin{cases} 1 & \text{si } I(x,y) > T \\ 0 & \text{si } I(x,y) \leq T \end{cases}$$



d) La squelettisation (amincissement) [3] :

Pour chaque pixel p_0 on considère son voisinage immédiat de 8 pixels $\{p_i, i \in [1..8]\}$. Pour l'algorithme de squelettisation on considère les définitions suivantes :

1. P_0 est un point frontière *Nord* si $P_2 = 0$

P ₁	P ₂	P ₃
P ₈	P	P ₄
P ₇	P ₆	P ₅

2. P_0 est un point frontière *Est* si $P_4 = 0$

3. P_0 est un point frontière *Sud* si $P_6 = 0$

4. P_0 est un point frontière *Ouest* si $P_8 = 0$

5. P_0 est un point *8-terminal* si un seul de ses voisins est noir ($\exists i \in [1..8], p_i = 1$), il s'agit d'une minutie de type terminaison

6. P_0 est un point *8-isolé* si aucun de ses voisins n'est noir ($\forall i \in [1..8], p_i = 0$)

7. P_0 est un point *8-simple* si la connexité de ses 8 voisins n'est pas altérée quand on le transforme en pixel blanc

La squelettisation consiste à répéter les opérations d'érosion suivantes jusqu'à ce que plus aucun pixel ne soit changé :

- **Etape 1 :** tous les pixels noirs vérifiant (1) et (7) et ne vérifiant pas (5) et (6) sont changés en pixels blancs (érosion des points frontières Nord)
- **Etape 2 :** tous les pixels noirs vérifiant (2) et (7) et ne vérifiant pas (5) et (6) sont changés en pixels blancs (érosion des points frontières Est)

- **Etape 3 :** tous les pixels noirs vérifiant (3) et (7) et ne vérifiant pas (5) et (6) sont changés en pixels blancs (érosion des points frontières Sud)
- **Etape 4 :** tous les pixels noirs vérifiant (4) et (7) et ne vérifiant pas (5) et (6) sont changés en pixels blancs (érosion des points frontières Ouest)



e) La détection des minuties :

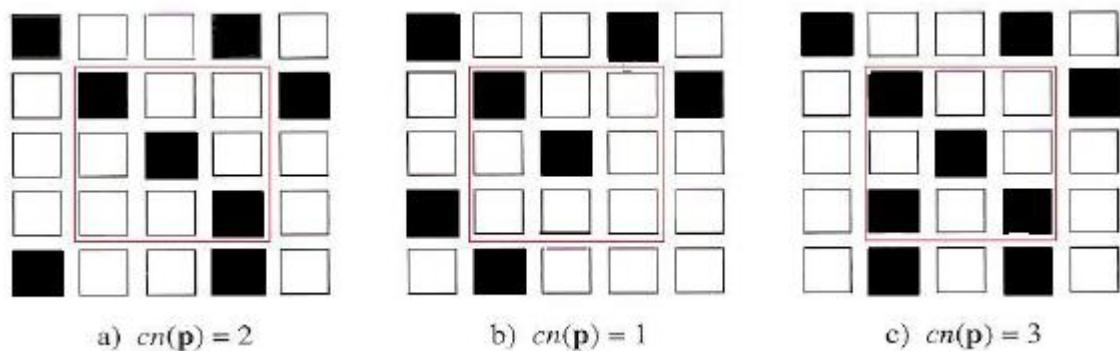
Une fois l'image est amincie, on procède à la détection des minuties par la technique du *crossing number*, initiée par Arcelli.

Le *crossing number* $cn(p)$ d'un pixel p se calcule par la formule suivante [4]:

$$cn(p) = \frac{1}{2} \sum_{i=1}^8 \left| val(p_{i \text{ mod } 8}) - val(p_{i-1}) \right|$$

p_0, p_1, \dots, p_7 sont les 8 pixels au voisinage de p et $val(p) \in \{0,1\}$ (car image binarisée). Alors, un pixel p dont $val(p) = 1$:

- * correspond à une terminaison si $cn(p) = 1$;
- * correspond à un point d'une ligne de l'empreinte si $cn(p) = 2$;
- * correspond à une bifurcation si $cn(p) \geq 3$.



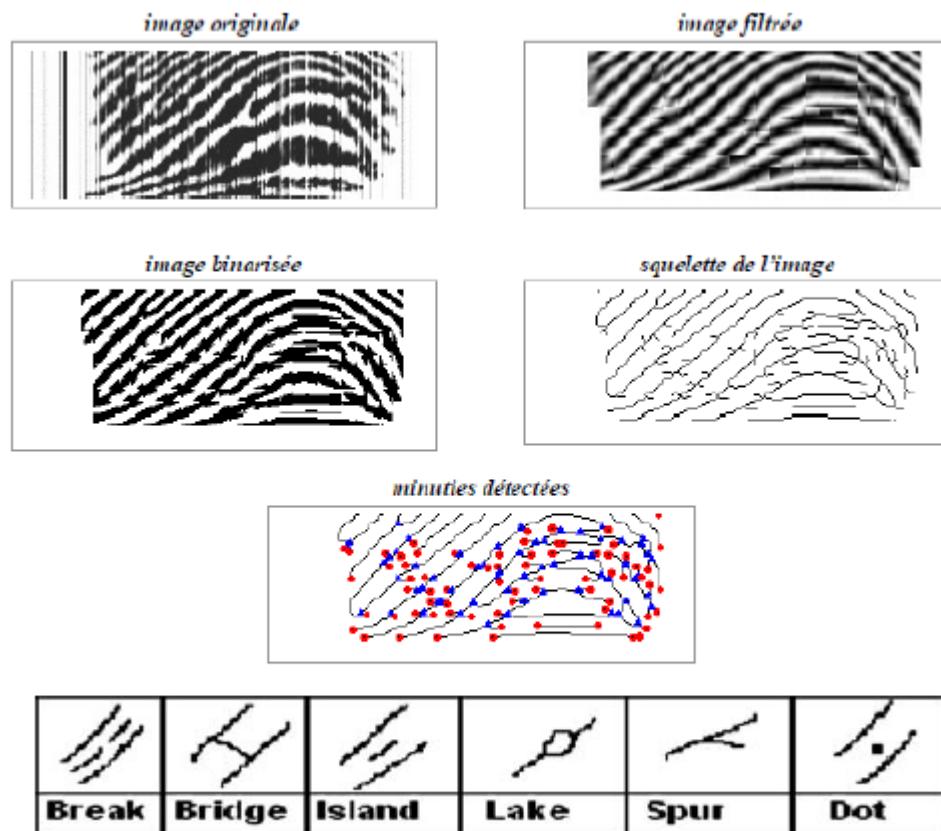
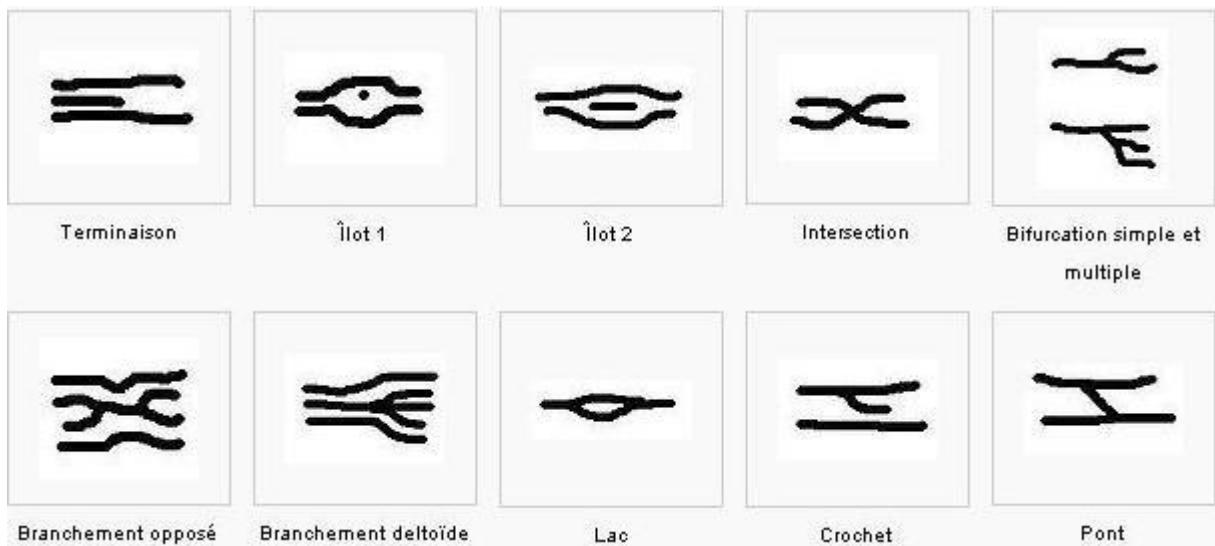


Figure : Exemple de détection de fausses minuties

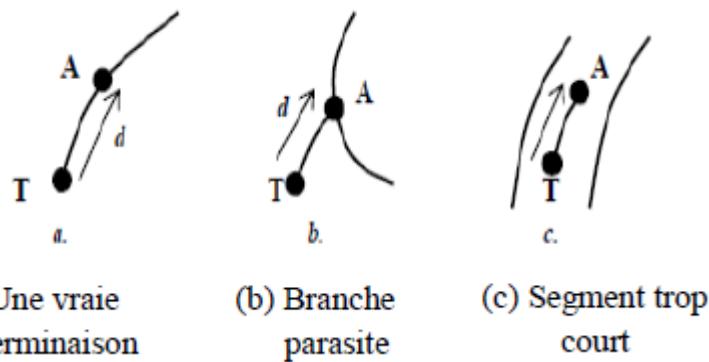


2.4 Post-traitement [3]:

L'étape d'extraction de minuties est souvent suivie par un post traitement qui essaye d'éliminer les fausses minuties.

→ Pour éliminer les fausses terminaisons, on suit les règles suivantes :

- 1- S'il existe un bloc adjacent au bloc contenant la terminaison $T(x_T, y_T)$ et appartenant au bord de l'image, alors T est une fausse minutie et on l'élimine.
- 2- Pour les terminaisons restantes T , on parcourt la strie qui lui est associée sur une distance maximum K_1 , jusqu'à atteindre le point A ($d = \widehat{TA} \leq K_1$).



→ Pour éliminer les fausses bifurcations, on suit les règles suivantes :

On parcourt les 3 stries qui sont associées à la bifurcation B sur une distance maximum de K_1 , jusqu'à atteindre 3 points A_1 , A_2 et A_3 .

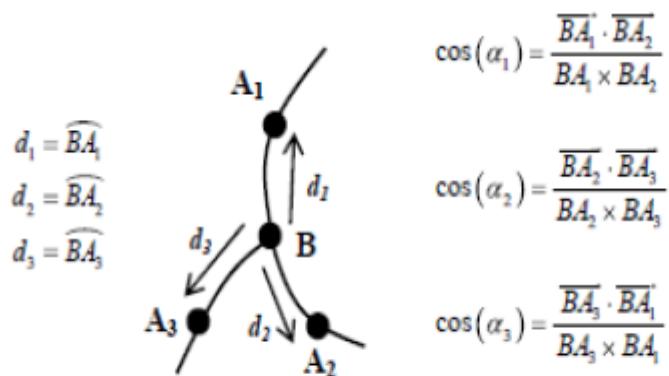


Figure : Définitions associées à une bifurcation lors de la phase de validation

- 1- Si $d_1 < k_1$, $d_2 < k_1$ et $d_3 < k_1$: la zone circulaire de centre B et de rayon k_1 contient au moins 4 minuties. On considère alors que nous sommes dans une zone très bruitée et que B est une fausse bifurcation.
- 2- Si $CN(A_1)=1$ ou $CN(A_2)=1$ ou $CN(A_3)=1$: au moins une des stries mène à une terminaison. On est dans le cas d'une branche parasite, le point B et les terminaisons atteintes ne sont pas validés.

3- Si $A_1 = A_2$ ou $A_2 = A_3$ ou $A_3 = A_1$: deux des stries mènent au même point. On est dans le cas d'un lac, le point B et la bifurcation atteinte ne sont pas validés.

- 4- Si nous avons 2 des stries qui mènent à 2 bifurcation A_1 et A_2 ($CN(A_1)=3$ et $CN(A_2)=3$). Dans ce cas on calcule la différence angulaire α_1 ainsi que la distance $\|A_1 A_2\|$ entre les deux bifurcations rencontrées. Si $|\cos(\alpha_1)| > \cos\frac{\pi}{4}$ et $\|A_1 A_2\| \leq \lambda$ alors on est dans le cas d'un triangle et on considère que B est une vraie bifurcation tandis que A_1 et A_2 sont des fausses.
- 5- Une seule des stries mène à une bifurcation A_1 ($CN(A_1)=3$). On calcule les différences angulaires α_2 et α_3 ainsi que la distance entre A_1 et B . Si $|\cos(\alpha_2)| < \cos\frac{\pi}{4}$, $|\cos(\alpha_3)| < \cos\frac{\pi}{4}$ et $\|BA_1\| \leq \lambda$ alors on est dans le cas d'un pont et A_1 et B sont considérées comme de fausses minuties.

2.5 Le fichier signature (gabarie) :

Le fichier signature correspond à l'information utile minimum contenue dans l'image qui est nécessaire à la reconnaissance et qu'il s'agit de la liste des minuties détectées.

- Pour chaque minutie détectée et validée on extrait 3 caractéristiques [3]:

- Le type de minutie : *bifurcation ou terminaison* (1 bit).
- La position de la minutie dans l'image : *coordonnées* (x, y) (2 octets).
- La direction du bloc local associé à la strie : θ (2 octets)

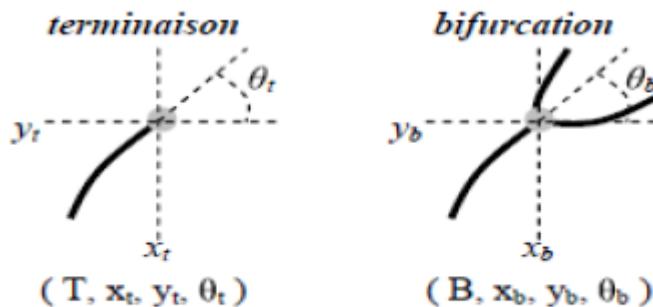


Figure : Les caractéristiques extraites d'une minutie

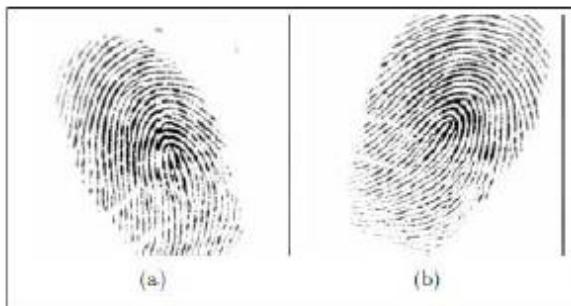
- Ensuite, les caractéristiques de chaque minutie seront reprises dans un vecteur. Donc, notre fichier comporte N vecteur de minutie comme suit :

$$S = \{M_i = (x_i, y_i, \theta_i, t_i) \mid i \in [1..N]\} \quad \|S\| = N$$

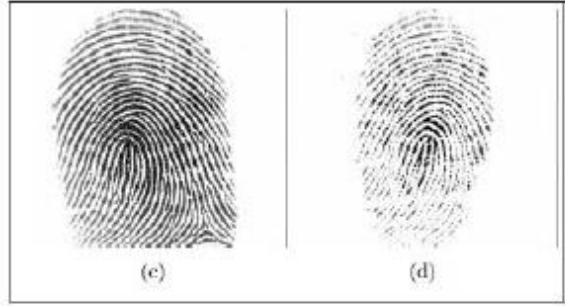
- Le fichier signature a une taille < 0.5 ko ce qui est un gain de mémoire par rapport à la taille d'image de 256 pixels (64 ko) et il sera sauvegardé dans une base de donnée.

3. La phase de reconnaissance de l'empreinte digitale (fingerprints matching)

Le problème de la phase de comparaison c'est qu'une même empreinte peut avoir plusieurs images à causes de plusieurs raisons :

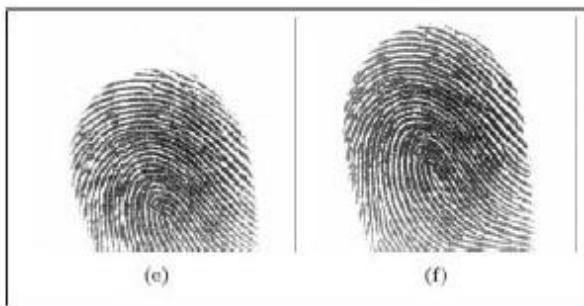


le doigt était penché à gauche sur le lecteur



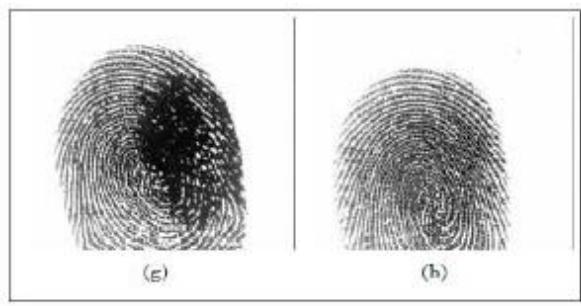
le doigt a été plus pressé

le doigt a été très légèrement pressé



une partie du doigt n'a pas été apposée sur le lecteur

le doigt a été correctement apposé sur le lecteur



le doigt était sale

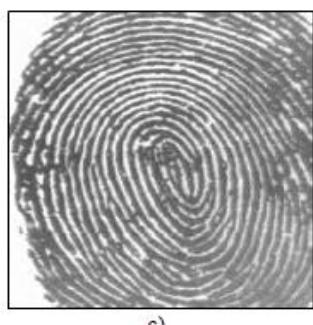
le doigt était propre



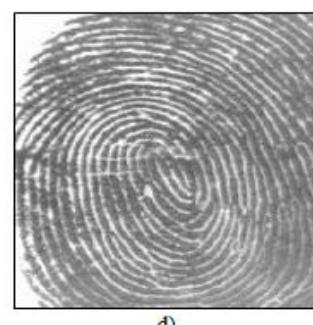
a)



b)

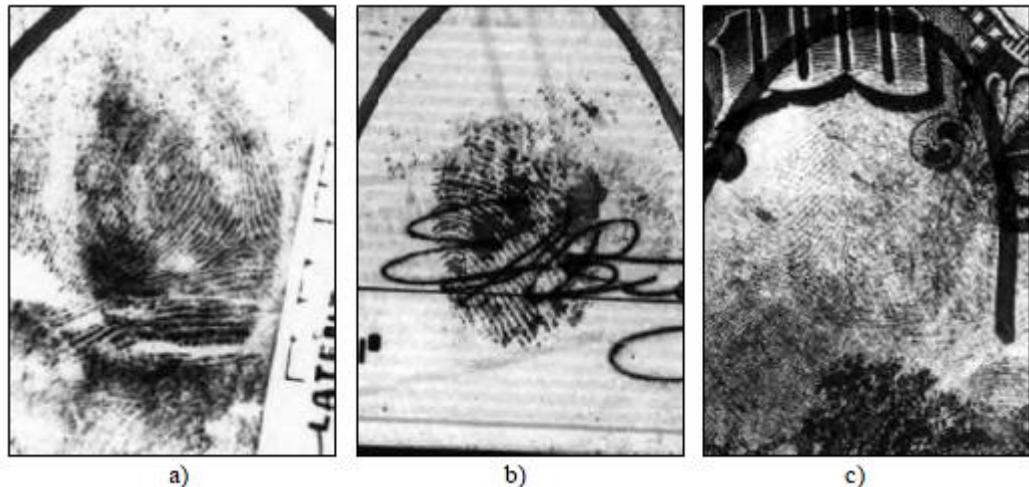


c)



d)

Figure : Difficulté de matching d'empreinte digitale. Images d'empreintes digitales en a) et b) apparaissent différentes à un œil non averti, mais ils ont pris du même doigt. Images d'empreintes digitales dans c) et d) apparaissent similaires à un œil non averti, mais ils sont de différents doigts



Les différentes approches de comparaison des empreintes digitales sont classées en trois grandes familles [1]:

- **Assortiment basé sur la corrélation** : elle consiste à comparer les matrices de pixels des images de 2 empreintes, et calculer la corrélation de ses pixels
- **Assortiment basé sur les rides** : dans cette approche, on utilise des caractéristiques extraites des rides (orientation, texture, forme de ride, ...) pour comparer les 2 empreintes
- **Assortiment basé sur minuties** : c'est l'approche la plus utilisée dans la littérature, elle se base sur des minuties qui sont extraites à partir des 2 empreintes digitales à comparer.

L'assortiment basé sur minuties [5] généralement appelée le “**matching de minuties**”, consiste à comparer les minuties de deux empreintes en les plaçant dans le plan bidimensionnel et à trouver les paires de minuties qui ont le même emplacement et la même orientation.

- On représente l'empreinte à identifier par **I**, et l'empreinte de la base de données par **T**.
On peut donc voir les deux empreintes à comparer **I** et **T** comme des vecteurs de minuties **m** :

$$\begin{aligned} \mathbf{T} &= \{m_1, m_2, \dots, m_m\} & \text{où} & & m_i &= \{x_i, y_i, \theta_i\} & i &= 1..m, \\ \mathbf{I} &= \{m'_1, m'_2, \dots, m'_n\} & & & m'_j &= \{x'_j, y'_j, \theta'_j\} & j &= 1..n. \end{aligned}$$

Où m et n sont respectivement le nombre de minuties de **T** et **I**.

- On considère qu'il y a un “match” entre une minutie m_i de **T** et une minutie m'_j de **I** si [5]:

et

$$sd\left(\mathbf{m}'_j, \mathbf{m}_i\right) = \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2} \leq r_o$$

$$dd\left(\mathbf{m}'_j, \mathbf{m}_i\right) = \min\left(|\theta'_j - \theta_i|, 360 - |\theta'_j - \theta_i|\right) \leq \theta_o.$$

Tel que : sd c'est la distance

dd c'est la différence

r_o et θ_o ce sont la tolérance et la tolérance angulaire respectivement

- Chaque minutie \mathbf{m}'_j de \mathbf{I} est donc comparée à toutes les minuties de \mathbf{T} pour trouver la plus semblable à \mathbf{m}'_j . Ce processus peut prendre un temps considérable pour une base de données contenant des milliers de Templates.

Référence

- [1] Raja Lehtihet. 'Exploitation des mécanismes rétino-corticaux pour la reconnaissance des images d'empreintes digitales'. Thèse de doctorat, université Constantine 2. 2014.
- [2] Benchennane Ibtissam. 'Etude et mise au point d'un procédé biométrique multimodale pour la reconnaissance des individus'. Thèse de doctorat, université d'Oran, 2016.
- [3] Selma Benayad, Habiba Bousri. 'La reconnaissance des empreintes digitales'. Mémoire de master, université de Jijel. 2011.
- [4] Benadjimi Noussaiba, Boubakeur Meriem Romaissa. 'Mise en œuvre d'une plateforme de test pour la sécurisation des données biométriques par la méthode du Fuzzy Vault'. Mémoire de fin d'étude pour l'obtention du diplôme d'ingénieur d'Etat, Ecole nationale Supérieure d'Informatique (ESI). 2015.
- [5] Souheila Benkhaira. 'Systèmes multimodaux pour l'identification et l'authentification biométrique'. Mémoire de magister, université de Skikda. 2010.
- [6] DANG Hoang Vu. 'Biométrie pour l'identification'. Institut de la Francophonie pour l'Informatique. 2005