

Analyse des documents graphiques :
une approche par reconstruction d'objets

15 juin 2007

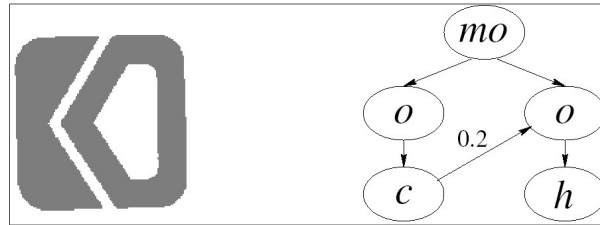
Introduction première partie

Comme présenté au cours du chapitre d'introduction, l'extraction de primitives graphiques est la première étape utilisée par les approches structurelles pour l'analyse des documents. Elle correspond à l'extraction de caractéristiques préalable à l'étape de reconnaissance dans une chaîne de traitements. Elle s'appuie sur différentes méthodes issues des applications d'interprétation de documents techniques [Ablameyko 00] et de reconnaissance de symboles [Lladós 02]. D'autres applications les exploitent également comme l'interprétation de documents structurés [Trupin 03], la reconnaissance de diagrammes (formules, partitions, ...) [Blostein 00], la reconnaissance de caractères (script [Suen 03] et latin [Vinciarelli 02]), ...

Les primitives graphiques extraites sont à base de vecteurs (vecteur, cercle, courbe, ...) et/ou de pixels (pixel, plage, région, ...) [Murray 96]. Elles sont par la suite agencées dans des structures de plus haut-niveau le plus souvent de type graphe¹ [Tombre 96]. Les graphes permettent la recherche de sous-graphes pour l'extraction de sous-structures dans un document graphique [Lladós 01] [Sánchez 02]. Ils sont donc particulièrement adaptés aux documents graphiques dans lesquels les parties graphiques sont en interconnection. La variabilité des documents graphiques implique que les graphes extraits sont le plus souvent bruités [Tombre 96]. Le problème d'exploitation des graphes par un système est donc un problème de graphes inexacts.

La figure suivante donne une représentation d'un symbole par un graphe de concavité [Badawy 02]. Dans ce graphe, les primitives graphiques représentent les composantes (o) et les occlusions (h) du symbole. Celles-ci ont été structurées selon leurs relations de composition et de concavité. Le noeud attribué (c) représente un attribut de concavité d'un noeud composante (o). Le noeud attribué (mo) représente lui un groupement d'objets (o). Ce graphe est orienté, la représentation (\uparrow) des arcs indiquent le sens de l'orientation. Parmi ces arcs, ceux non attribués correspondent à des relations de composition entre noeuds, et celui attribué (0.2) une relation d'emboîtement entre deux objets (o) via le noeud attribut de concavité (c).

¹Nous reportons le lecteur à l'Annexe A pour une introduction sur les graphes.



Graphe de concavité d'un symbole

Dans cette première partie, nous présentons notre étude bibliographique en matière d'extraction de primitives graphiques et de gestion des connaissances. Nous présentons dans le chapitre (1) un état de l'art sur l'extraction de primitives graphiques. Nous y détaillons les différentes méthodes d'extraction employées dans la littérature. Nous illustrons les différents niveaux d'extraction selon lesquels se décomposent ces méthodes. Nous comparons également les avantages et les inconvénients de chacune des méthodes ainsi que leur complémentarité. Nous concluons alors sur l'intérêt de la combinaison des différentes méthodes pour l'extraction des primitives graphiques. Cette combinaison soulève cependant le problème de l'échange des primitives graphiques entre les différentes méthodes. Plus largement, elle soulève celui de la gestion des connaissances dites graphiques au sein des systèmes d'analyse des documents. Nous abordons cette problématique dans un état de l'art au cours du chapitre (??). Nous l'avons plus particulièrement centré sur les problèmes de représentation (formalismes et modèles) des connaissances graphiques. Nous montrons que les systèmes manipulent des connaissances graphiques proches mais représentées de différentes façons en particulier en ce qui concerne les modèles. Nous en concluons que le problème de l'échange des connaissances graphiques entre opérateurs d'extraction est avant tout un problème d'interopérabilité sur les modèles. Nous argumentons ensuite que cette interopérabilité ne peut être résolue que par une approche à base de multi-représentation. Nous proposons alors de formaliser la combinaison des opérateurs d'extraction comme un processus de reconstruction d'objets.

Chapitre 1

Extraction de primitives graphiques, état de l'art

1.1 Introduction

Les méthodes d'extraction de primitives graphiques sont communément utilisées dans la littérature. Elles se décomposent en différents niveaux d'extraction selon que les primitives graphiques d'entrée soient à base de vecteurs et/ou de pixels. Celles basées sur des primitives à base de vecteurs¹ sont de diverses natures : approximation mathématique (vecteur, arc, et courbe) [Rosin 97], appariement de contours [Ramel 96], recherche d'intersections [Balaban 95], détection de polygones [Ferreira 03], . . . Celles basées sur les primitives à base de pixels sont plus communément répandues. Différents états de l'art y sont spécifiquement consacrés : [Loncaric 98], [Wenyin 99], [Tombre 00], et [Song 03]. Les images de document graphique véhiculent généralement des informations de type fond/forme. Les méthodes d'extraction exploitent cette propriété [Ablameyko 00], elles sont majoritairement conçues pour le traitement des images binaires. Celles-ci sont en effet plus simples à traiter et à manipuler. Lorsque les images traitées ne sont pas binaires, ou lorsqu'elles sont dégradées, on procède alors à leur pré-traitement². Suite au pré-traitement différentes familles de méthodes sont utilisées pour extraire les primitives graphiques. Nous en avons répertorié sept que nous présentons sur la figure (1.1) : les méthodes à base de détection de contours (a), squelettisation (b), parcours de formes (c), décomposition en plages (d), segmentation en régions (e), sous-échantillonnage (f), et graphe de composantes (g). Dans ce chapitre nous présentons ces différentes familles dans les sections (1.2) à (1.8). Dans la section (1.9) nous comparons les avantages et les inconvénients de chacune des méthodes ainsi que leur complémentarité. Finalement, dans la section (1.10) nous concluons.

¹Nous n'aborderons pas ces méthodes dans ce manuscrit.

²Nous reportons le lecteur à [Ablameyko 00] sur ces aspects.

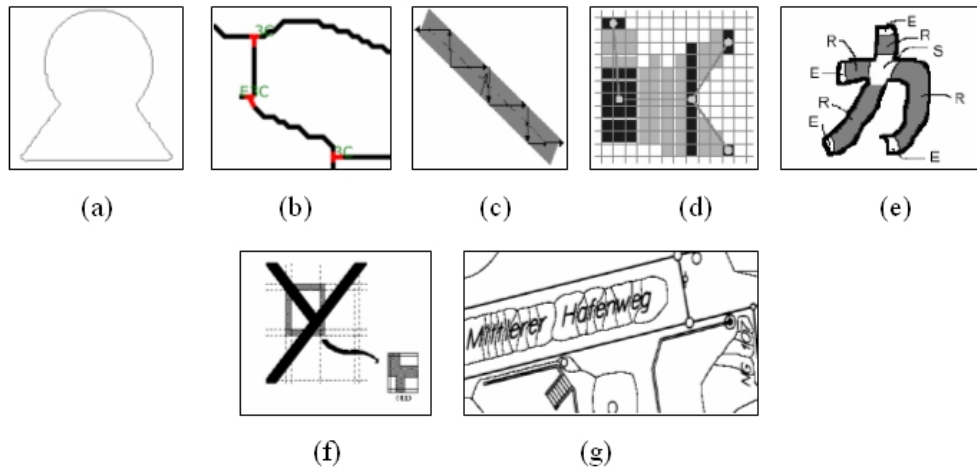


FIG. 1.1 – (a) détection de contours (b) squelettisation (c) parcours de formes
 (d) décomposition en pages (e) segmentation en régions
 (f) sous-échantillonnage (g) graphe de composantes

1.2 Les méthodes à base de détection de contours

Les méthodes à base de détection de contours sont couramment utilisées en analyse d'image de document graphique [Ablameyko 00]. Elles constituent l'une des deux familles de méthodes les plus utilisées avec celles à base de squelettisation [Tombre 00]. Elles se basent sur l'analyse des contours des formes d'un document pour extraire des primitives graphiques. La figure (1.2) donne un exemple de résultat de détection de contours [Hasan 00]. On distingue alors diverses catégories de méthodes [Ablameyko 00] : par détection morphologique des contours, par suivi de contours et par balayage de lignes. Celles par détection morphologique utilisent deux étapes [Song 02a], une d'extraction des images de représentation des contours et une de chaînage de ces images. Les autres réalisent la détection et le chaînage des contours en une étape. Nous les présentons chacune de ces catégories de méthodes par la suite.

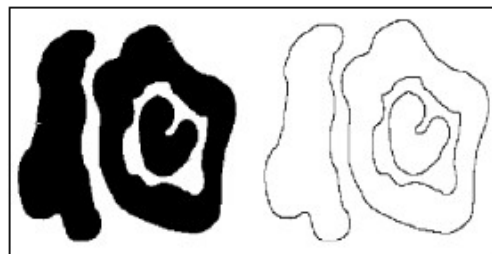


FIG. 1.2 – Exemple de détection de contours

Les méthodes à base de détection morphologique [Zhang 98] [Hasan 00] appliquent des opérations de morphologie mathématique à l'image [Coster 89]. Ces opérations sont largement utilisées en analyse d'image de document, en particulier pour le pré-traitement [Kasturi 02]. Elles utilisent des opérations logiques (*et* (\cap), *ou* (\cup)) entre les pixels (p) de l'image et leurs voisinages (P_n) afin de déterminer l'intensité du pixel (p') de l'image résultat (équation (1.1)). L'opération teste le voisinage d'un pixel (p) à l'aide d'un élément structurant (ou masque) (E) le plus souvent en croix ($+$)³. Ces opérations de morphologie appliquent les éléments structurants globalement à l'image (I). On note (\ominus) l'opération d'érosion basée sur le (*et*) logique, et (\oplus) l'opération de dilatation basée sur le (*ou*) logique (équation (1.2)). Ces opérations permettent d'obtenir respectivement les images érodée (I_e) et dilatée (I_d) à partir d'une image (I). À partir de ces deux opérations, les équations (1.3) rappellent celles d'ouverture et de fermeture permettant d'obtenir respectivement les images (I_o) et (I_f). La figure (1.3) donne des résultats d'opérations morphologiques (b-e) à partir d'une image (a) [Ablameyko 00] basées sur un élément structurant ($+$)³.

$$P_n = \begin{array}{|c|c|c|} \hline p_3 & p_2 & p_1 \\ \hline p_4 & p & p_0 \\ \hline p_5 & p_6 & p_7 \\ \hline \end{array} \quad E = \begin{array}{|c|c|c|} \hline * & 1 & * \\ \hline 1 & 1 & 1 \\ \hline * & 1 & * \\ \hline \end{array} \quad \begin{array}{l} p' = p \cap (p_0 \cap p_2 \cap p_4 \cap p_6) \\ p' = p \cup (p_0 \cup p_2 \cup p_4 \cup p_6) \end{array} \quad (1.1)$$

$$I_e = I \ominus E \quad I_d = I \oplus E \quad (1.2)$$

$$I_o = (I \ominus E) \oplus E \quad I_f = (I \oplus E) \ominus E \quad (1.3)$$

$$I_1 \setminus I_2 = (I_1 \cap \bar{I}_2) \cup (\bar{I}_1 \cap I_2) \quad \partial_e = I \setminus I_d \quad \partial_i = I \setminus I_e \quad (1.4)$$

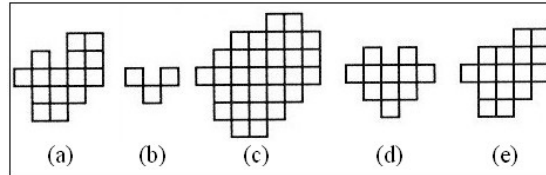


FIG. 1.3 – Exemples de résultats d'opérations morphologiques (a) image (b) érosion (c) dilatation (d) ouverture (f) fermeture

À partir des opérations de morphologie mathématique (équations (1.2) et (1.3)) [Hasan 00] définit les opérations de détection morphologique des contours externes (∂_e) et internes (∂_i) d'une image (équations (1.4)). Dans ces équations, le (\setminus) correspond à une différence logique (*ou exclusif*). Ces contours externes et internes⁴ correspondent respectivement aux pixels contours du fond connectés à la composante, et de la forme connectés au fond entourant la composante [Hasan 00]. La figure (1.4) illustre ces deux types de contours (b) (c) à partir d'une image (a).

³ 4-voisinage selon les axes Nord-Sud Ouest-Est.

⁴ Termes polysémiques, [Ramel 96] [Chang 04] désignent ainsi les contours englobants/englobés.

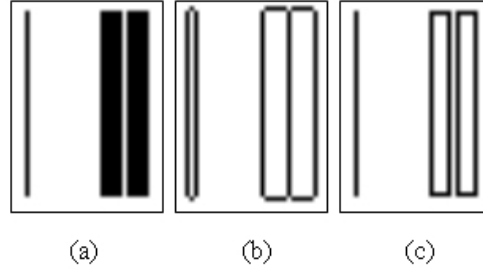


FIG. 1.4 – Détection morphologique des contours internes et externes
 (a) image (b) contours externes (c) contours internes

La détection morphologique des contours est généralement étendue à la détection par test de voisinage [Rosenfeld 82]. Il s'agit de concaténer les opérations de détection morphologique des contours (équations (1.4)) afin d'effectuer qu'une seule passe sur l'image. [Rosenfeld 82] définit l'affectation d'un pixel contour externe (p_e) de l'image des contours à partir d'un test logique entre le pixel (p) de l'image initiale avec son voisinage (P_n) (équation (1.5)). Cette détection par test de voisinage est également extensible à la détection de contours internes (p_i). Plusieurs systèmes privilégient cette méthode afin de réduire la complexité de traitement de la détection de contours [Han 94] [Labelle 98].

$$P_n = \begin{array}{|c|c|c|} \hline p_3 & p_2 & p_1 \\ \hline p_4 & p & p_0 \\ \hline p_5 & p_6 & p_7 \\ \hline \end{array} \quad \begin{array}{l} p_e = \bar{p} \cap (p_0 \cup p_2 \cup p_4 \cup p_6) \\ p_i = p \cap (\bar{p}_0 \cup \bar{p}_2 \cup \bar{p}_4 \cup \bar{p}_6) \end{array} \quad (1.5)$$

Les méthodes à base de détection morphologique des contours permettent donc de produire des images de représentation des contours. Afin d'extraire les primitives graphiques de type contour, les systèmes basés sur la détection morphologique [Han 94] [Labelle 98] [Zhang 98] [Hasan 00] exploitent une deuxième étape dite de chaînage [Nunes 00]. Cette dernière est utilisée afin de structurer les pixels contours en chaînes de points [Hasan 00]. La figure (1.5) illustre le principe du chaînage [Ah-Soon 98]. Celui-ci se fait par analyse du voisinage selon les 8 directions [Freeman 61]. Cette analyse est le plus généralement de trois types [Adam 98] : trigonométrique (droite à gauche), anti-trigonométrique (gauche à droite), et oscillatoire (alternativement trigonométrique et anti-trigonométrique). Les critères d'arrêt du chaînage sont de deux types [Nunes 00] : la rencontre du point de départ de chaînage (boucle), et l'absence de point (contour non fermé). Une chaîne de points ainsi extraite est ensuite encodée [Nunes 00] sous la forme d'une liste de coordonnées des points consécutifs ($x_i; y_i$), et sous la forme d'une liste de directions à partir du point d'entrée⁵ (figure (1.5)).

⁵Differential Chain Code (DCC) [Nunes 00]

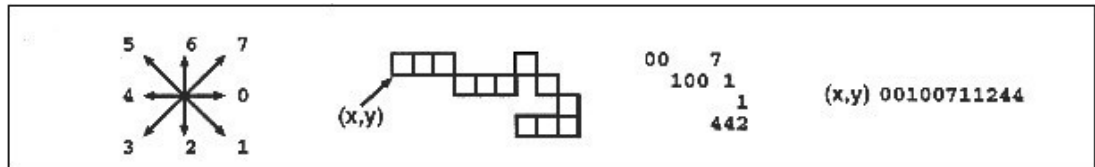


FIG. 1.5 – Principe de l'étape de chaînage

La détection morphologique de contours est donc basée sur des images de représentation des contours. Dans ces images, il est impossible pour un pixel contour d'appartenir simultanément à différents contours. Cependant, ce cas peut se produire fréquemment dans le cas de contours proches (figure (1.6) (a)). Il devient alors difficile de chaîner les contours à partir de ces images (figure (1.6) (b) et (c)). C'est pour cette raison que différents systèmes ([Ablameyko 93], [Lin 95], [Ramel 96], [Bishnu 99], [Yu 01], ...) exploitent des méthodes [Ablameyko 00] procédant à la détection et au chaînage des contours en une étape [Song 02a]. Ces méthodes sont proches des méthodes de marquage de composantes connexes⁶ et opèrent [Ablameyko 00] par suivi de contours ou par balayage de lignes.

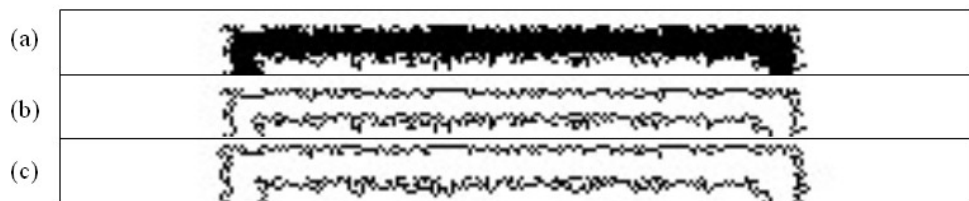


FIG. 1.6 – Détection morphologique de contours proches
 (a) image (b) contours internes (c) contours externes

Les méthodes de suivi de contours⁷ sont largement utilisées par différents systèmes : [Braun 95], [Turner 96], [Ramel 96], [Jordan 97], [Bishnu 99], [Yu 01], ... Malgré les variantes entre les méthodes de ces systèmes, celles-ci sont proches [Ablameyko 00] de la méthode initiale de [Black 81]. [Ramel 96] en illustre le principe à travers la figure (1.7) (a). Dans cette figure, (E) représente la forme dont les contours sont suivis, (P_a) le point de départ, et (P_c) le point parcouru. Le principe de parcours du contour est alors proche de celui d'une étape de chaînage. Ce parcours analyse le voisinage du pixel courant afin de rester sur les bords de la forme sans y pénétrer. Dans son état de l'art, [Turner 96] propose une taxinomie de ces méthodes selon que les contours soient internes ou externes, et selon que le parcours soit en 4 ou 8 connexités.

⁶Nous présentons ces méthodes page 36.

⁷contouring by line following [Ablameyko 00]

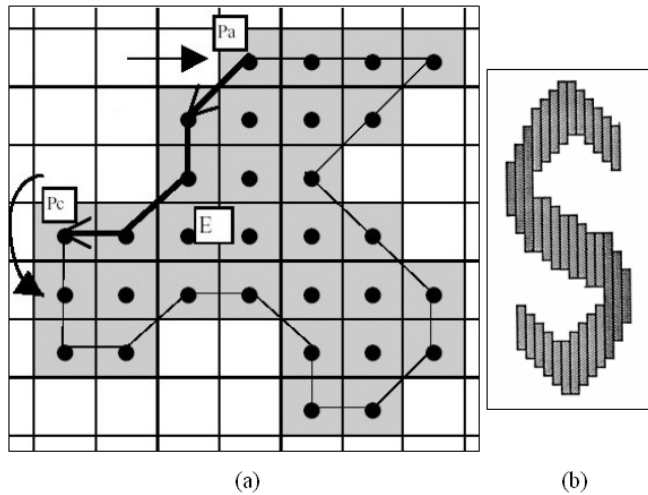


FIG. 1.7 – (a) suivi de contours (b) balayage de lignes

Les méthodes de détection de contours par balayage de lignes⁸ constituent une alternative aux méthodes de suivi de contours [Ablameyko 00]. Elles sont moins répandues, quelques systèmes les emploient [Ablameyko 93] [Lin 95] [Fan 98]. Elles sont basées sur l'encodage des lignes de l'image en plages⁹. Une plage est une séquence horizontale ou verticale de pixels noirs (de la forme) connectés. La figure (1.7) (b) donne un exemple de décomposition en plages verticales d'un caractère [Fan 98]. L'analyse de la configuration des plages (adjacence locale des plages et comparaison des longueurs) permet d'extraire les chaînes de points contours externes et internes de l'image.

Il n'existe pas travaux comparatifs sur les méthodes de détection de contours [Ablameyko 00]. Selon notre point de vue les méthodes par suivi de contours et par balayage de lignes permettent une meilleure détection que celles basées sur la morphologie mathématique. En effet elles n'utilisent pas d'image de représentation intermédiaire ce qui les rend moins complexes. Elles procèdent de plus à la détection et au chaînage des contours en une étape permettant de traiter ainsi les contours proches. C'est pour ces raisons qu'elles sont majoritairement utilisées dans la littérature [Ablameyko 93], [Lin 95], [Ramel 96], [Bishnu 99], [Yu 01], ... La comparaison entre les méthodes par suivi de contours et par balayage de lignes semblent moins évidente. Celles par balayage de lignes semblent plus complexes en temps de traitement et de mise en oeuvre pour un résultat d'extraction comparable [Ablameyko 00]. Leur exploitation est motivée [Lin 95] [Fan 98] par la construction a posteriori de graphes de plages⁹. Basé sur ces deux familles de méthodes les contours semblent robustes et simples à extraire [Wenyin 99]. Ils ne permettent pas cependant la représentation des jonctions des composantes ce qui en fait des primitives graphiques faiblement informative.

⁸contouring by scan line [Ablameyko 00]

⁹Nous abordons les méthodes à base de décomposition en plages page 26.

1.3 Les méthodes à base de squelettisation

1.3.1 Introduction

Les méthodes à base de squelettisation constituent la deuxième grande famille de méthodes pour l'extraction de primitives graphiques avec celles à base de détection de contours [Tombre 00]. Plusieurs états de l'art y sont en majeure partie consacrés [Wenyin 99] [Tombre 00] [Song 03]. Au même titre que les méthodes à base de détection morphologique des contours, elles se déroulent en deux étapes [Song 02a] : la squelettisation et la construction du graphe de squelette. Nous les présentons dans la suite de cette section.

1.3.2 La squelettisation

La squelettisation¹⁰ constitue la première étape de ces méthodes [Tombre 00]. Elle a été introduite dans les années 60 par [Blum 67]. Elle a depuis lors largement¹¹ été utilisée dans les systèmes d'OCR [Nagy 92]. Plusieurs états de l'art et travaux d'évaluation y sont spécifiquement consacrés : [Lam 92], [Zhang 93], [Lam 95], [Cowell 01], [Klette 03]. Un traitement implémentant une méthode de squelettisation est qualifiée de squelettiseur dans la "littérature française" [Taconet 90]. Un squelettiseur a pour but de produire un squelette à partir d'une image initiale. [Lam 92] propose la définition (1) suivante du terme squelette, la figure (1.8) en donne un exemple. La squelettisation procède par deux approches dans la littérature [Ablameyko 00] : par amincissements successifs [Lam 92] ou par transformée de distance [Klette 03]. Nous présentons chacune d'entre elles dans la suite de cette sous-section.

Définition 1 *Le terme "squelette" a été utilisé en général pour désigner une représentation d'une forme par un ensemble d'arcs et de courbes squelettiques (ou très minces)*¹².

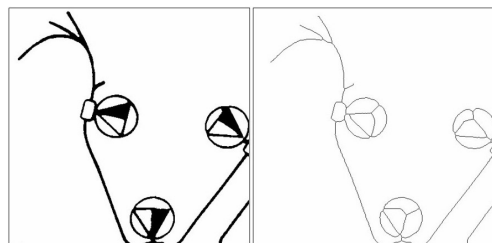


FIG. 1.8 – Squelette d'un extrait de plan de réseau

¹⁰thinning or skeletonization [Lam 92]

¹¹[Lam 92] comptabilise environ 300 articles "anglais" sur ce sujet au début des années 1990.

¹²The term "skeleton" has been used in general to denote a representation of a pattern by a collection of thin (or nearly thin) arcs and curves.

La squelettisation par amincissements successifs¹³ se base sur des suppressions successives des pixels contours de l'image jusqu'à obtenir l'image du squelette [Lam 92]. Les algorithmes de de squelettisation par amincissements successifs emploient pour cela une succession de passe ou itération¹⁴. La figure (1.9) illustre le principe de ces algorithmes via une squelettisation en 3 itérations [Ah-Soon 98].

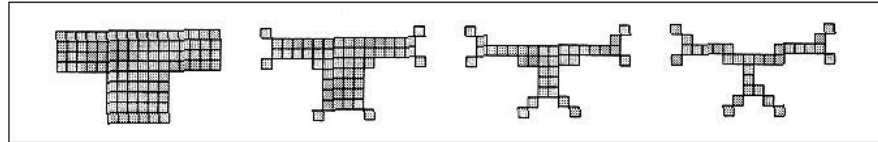


FIG. 1.9 – Squelettisation par amincissements successifs en 3 itérations

[Lam 92] distingue les algorithmes dits séquentiels de ceux dits parallèles. Dans les algorithmes séquentiels, les résultats d'une passe en cours (n) sont calculés à partir des résultats des ($n - 1$) passes précédentes, mais aussi des résultats de la passe (n) en cours. À l'opposé, les algorithmes parallèles n'utilisent durant la passe (n) que les résultats des ($n - 1$) passes précédentes [Cowell 01]. Pour cette raison, les algorithmes parallèles peuvent être utilisés sur des architectures parallèles [Montoya 98], ce qui optimise considérablement les temps de traitement. Les travaux de recherche privilégient d'ailleurs, depuis le début des années 1990, ces algorithmes au détriment des algorithmes séquentiels [Lam 95] [Lam 92]¹⁵.

Les algorithmes parallèles¹⁶ de squelettisation par amincissements successifs exploitent différents éléments structurants le plus souvent de type (3×3) [Lam 92]. Le voisinage induit par un élément structurant permet de définir en fonction de tests logiques si le pixel central de l'élément doit être supprimé ou non [Cowell 01]. Chaque application d'un élément structurant aux pixels contours de l'image correspond à une sous-passe ou sous-itération¹⁷ d'un algorithme. [Lam 92] identifie ainsi les algorithmes exploitant 1, 2, et 4 sous-itération(s). Cette décomposition permet de considérer seulement un sous-ensemble de pixels contours à chacune des sous-itérations de l'algorithme. En effet, l'analyse de l'ensemble des pixels contours en une seule passe peut provoquer une perte de connexité du squelette [Cowell 01]. Aussi, la pratique habituelle est donc de décomposer cette analyse via les sous-itérations [Lam 92]. Le choix entre les algorithmes basés sur 1, 2, et 4 sous-itération(s) dépend du compromis rapidité/précision souhaité [Lam 95]. Plus le nombre d'itérations est important, plus la squelettisation sera précise. Durant une sous-itération, après avoir marqué l'ensemble des pixels contours répondant aux critères de suppression, ceux-ci sont supprimés avant le démarrage de la sous-itération suivante [Lam 92].

¹³iterative shrinking [Ablameyko 00]

¹⁴cycle or iteration [Lam 92]

¹⁵Nous reportons le lecteur à [Lam 92] pour une présentation des algorithmes séquentiels.

¹⁶Nous omettrons ce terme par la suite.

¹⁷sub-cycle or sub-iteration [Lam 92]

Afin d'illustrer la mise en oeuvre des éléments structurants, nous présentons ceux utilisés dans l'algorithme de squelettisation par amincissements successifs de [Arcelli 75]. Celui-ci utilise différents éléments structurants (E_{12}) (E_{12}) (E_{21}) (E_{22}) afin d'effectuer différents tests logiques sur le voisinage des pixels contours (équations (1.6) et (1.7)). Les éléments structurants (E_{12}) et (E_{22}) correspondent respectivement aux rotations de ($\pi/2$) des éléments structurants (E_{11}) et (E_{21}). Dans ces éléments structurants, la présence de (1) (forme) et de (0) (fond) signifie que le pixel contour testé sera supprimé si ces (1) et (0) apparaissent aux emplacements correspondants dans son voisinage. Il s'agit donc de tests logiques d'égalité membre à membre entre l'élément structurant et le voisinage du pixel contour. Les astérisques (*) signifient que le pixel voisin à cet emplacement n'a pas d'incidence sur le test de suppression. Chacun des éléments structurants est utilisé dans chacune des quatre sous-itérations de l'algorithme pour la suppression d'un sous-ensemble de pixels contours dans l'ordre (E_{11}) (E_{21}) (E_{12}) (E_{22}).

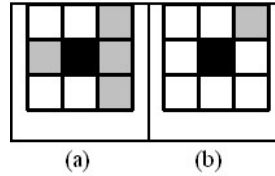
$$E_{11} = \begin{array}{|c|c|c|} \hline 0 & 0 & * \\ \hline 0 & 1 & 1 \\ \hline * & 1 & * \\ \hline \end{array} \quad E_{12} = \begin{array}{|c|c|c|} \hline * & 1 & * \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & * \\ \hline \end{array} \quad (1.6)$$

$$E_{21} = \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline * & 1 & * \\ \hline 1 & 1 & * \\ \hline \end{array} \quad E_{22} = \begin{array}{|c|c|c|} \hline 0 & * & * \\ \hline 0 & 1 & 1 \\ \hline 0 & * & 1 \\ \hline \end{array} \quad (1.7)$$

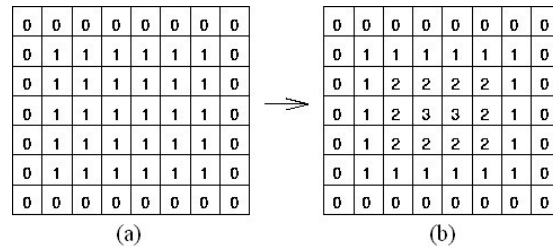
L'utilisation d'éléments structurants (3×3) par les algorithmes de squelettisation par amincissements successifs permet de multiples configurations [Cowell 01]. En effet, considérant les 8 coefficients voisins $\{c_0, c_7\}$ du coefficient central (c) d'un élément structurant on a $i_m = 256$ (E_i) configurables selon que l'on considère de 0 à 8 coefficients voisins (équation (1.8)). Cependant, en raison de la symétrie croisée des éléments structurants (3×3), seuls 64 (E_i) sont configurables. Enfin, parmi ces 64 (E_i) seuls ($i'_m = 32$) répondent aux conditions (1) et (2) suivantes de suppression des pixels contours. Considérant que (s) éléments structurants différents successifs sont utilisés durant les sous-itérations d'un algorithme, le nombre (C_E) de configurations devient alors conséquent (équation (1.8)). C'est pour cette raison que les éléments structurants utilisés dans ces algorithmes sont déterminés de façon empirique [Lam 92].

$$E_i = \begin{array}{|c|c|c|} \hline c_3 & c_2 & c_1 \\ \hline c_4 & c & c_0 \\ \hline c_5 & c_6 & c_7 \\ \hline \end{array} \quad i_m = \sum_{k=0}^8 C_8^k = 256 \quad C_E = A_{i'_m}^s \quad (1.8)$$

1. La suppression du pixel contour central ne doit pas entraîner de ruptures de connexions des autres pixels voisins dans l'élément structurant (figure (1.10) (a)).
2. La suppression du pixel contour central ne doit pas laisser moins de deux autres pixels voisins dans l'élément structurant (figure (1.10) (b)).

FIG. 1.10 – Éléments structurants non valides¹⁸

Les algorithmes de squelettisation par amincissements successifs garantissent ainsi un squelette exact [Lam 92] : le squelette est connexe et composé d'arcs et de courbes d'épaisseur unitaire (1 pixel). Ils possèdent cependant trois inconvénients majeurs [Chen 96] : ils sont réputés pour des temps de calcul longs, ils sont sensibles aux bruits et à l'épaisseur des traits et ils ne conservent pas d'information sur l'épaisseur. Une autre approche de squelettisation [Klette 03] permet de pallier en partie ces inconvénients. Elle est basée sur les transformées de distance¹⁹. Une transformée de distance est une opération appliquée à une image binaire. Le résultat d'une transformation est une image de topologie identique, mais dont l'intensité d'un pixel de la forme correspond à la distance le séparant de son pixel fond le plus proche [Smith 04]. On parle alors de carte des distances. La figure (1.11) donne un exemple d'image et de carte des distances résultat de la transformée.

FIG. 1.11 – Exemple de résultat de transformée de distance
(a) image (b) carte des distances

Il existe différentes transformées de distance [Klette 03] [Smith 04]. Elles sont cependant comparables à la transformée initiale de distance de chanfrein²⁰. Cette distance a été introduite par [Barrow 77]. Elle est largement utilisée en analyse d'image, différents états de l'art lui sont spécifiquement consacrés [Nacken 93] [Thiel 01] [Ronse 04]. Afin d'introduire cette distance, les équations (1.9) et (1.10) rappellent la définition d'une distance dans l'espace discret (Z^2) et ses axiomes d'égalité de symétrie et d'inégalité triangulaire. À partir de ces définitions, l'équation (1.11) rappelle les distances de 4-connexité (d_4) et 8-connexité (d_8)²¹.

¹⁸ Les éléments (noirs/gris) (blancs) correspondent respectivement aux (1) (0) (équations (1.6), (1.7))

¹⁹ distance transforms [Smith 04]

²⁰ chamfer distance [Nacken 93]

²¹ Manhattan and Cheeseboard distances [Nacken 93]

$$(x, y) \in Z \quad p(x_p, y_p) \wedge q(x_q, y_q) \in Z \quad d(p, q) \in Z^2 \quad (1.9)$$

$$d(p, q) = 0 \Leftrightarrow p = q \quad d(p, q) = d(q, p) \quad d(p, q) + d(q, r) \leq d(p, r) \quad (1.10)$$

$$d_4(p, q) = |x_p - x_q| + |y_p - y_q| \quad d_8(p, q) = \max(|x_p - x_q|, |y_p - y_q|) \quad (1.11)$$

Le calcul de la distance de chanfrein consiste à considérer l'ensemble des pixels de l'image et leurs 8-connexités comme un 8-graphe²². La figure (1.12) (a) illustre un 8-graphe d'une image de dimension (5×7) pixels. Dans ce graphe, les noeuds $\{\bullet\}$ représentent les pixels, et les arcs leurs relations de 8-connexités. La distance de chanfrein est ensuite calculée par pondération de ce 8-graphe. La figure (1.12) (b) illustre les relations de pondération, entre un pixel et son 8-voisinage, utilisées pour le calcul de la distance de chanfrein. La distance de pondération ($d_p(p, p_n)$) entre le pixel central (p) et un voisin (p_n) correspond à un coefficient de pondération $\{a \vee b\}$. Cette distance respecte les trois axiomes définis précédemment (équation (1.10)). Ces coefficients de pondération sont définis selon un masque le plus généralement de type (3×3) (figure (1.12) (c)). Ce masque (3×3) est alors qualifié de masque de chanfrein²³, et répond à une structure symétrique (a, b) tel que $(a > 0)$ et $(a < b \leq 2 \times a)$ ²⁴. Le type de distance (d_p) dépend du masque de chanfrein utilisé. On a la distance en 4-connexité (d_4) et 8-connexité (d_8) (équation (1.11)) pour respectivement des masques de types $(a = 1, b = 2)$ et $(a = 1, b = 1)$. Il existe également différentes variantes de masques de types $(\alpha \times d_4 + \beta \times d_8)$ ²⁵. Considérant deux pixels (q_0) et (q_1) à un emplacement quelconque dans le 8-graphe pondéré, la distance de chanfrein (d_c) entre ces deux pixels correspond à la somme du chemin de poids minimum et de longueur (l) (équation (1.12)).

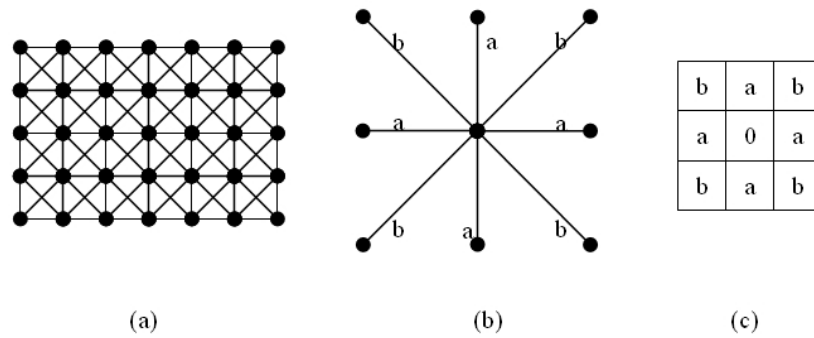


FIG. 1.12 – 8-graphe pondéré par masque de chanfrein (3×3)
(a) 8-graphe (b) relations de pondération au 8-voisinage (c) masque de chanfrein

²²Nous reportons le lecteur à l'Annexe A pour une introduction sur les graphes.

²³Nous reportons le lecteur à [Nacken 93] pour une présentation des masques (5×5) .

²⁴Nous reportons le lecteur à [Thiel 01] pour une description des propriétés du masque de chanfrein.

²⁵[Baja 94b] utilise par exemple un masque $(3, 4)$ de type $(1 \times d_4 + 2 \times d_8)$

$$d_c(q_0, q_l) = \sum_{i=0}^{l-1} d_p(q_i, q_{i+1}) \quad (1.12)$$

L'approche la plus naïve pour calculer $(d_c(q_0, q_l))$ serait de l'évaluer pour tout $(q_i \neq 0 \ q_j \neq l \in Z^2)$, ce qui donnerait une complexité proportionnelle à $(Z^2 \times Z^2)$. Lorsque le masque de chanfrein respecte les conditions évoquées précédemment²⁴, il existe un algorithme séquentiel²⁶ en deux passes permettant ce calcul en une complexité $(Z^2 \times m)$, ou (m) correspond à la taille du masque de chanfrein (généralement 3). Dans cet algorithme séquentiel, les résultats de la passe en cours par un masque sont donc pris en compte ultérieurement au cours de cette même passe. Le masque de chanfrein est décomposé en deux sous-masques pour chacune des passes dits masque antérieur et masque postérieur. L'équation (1.13) donne l'exemple du masque (M) de chanfrein $(1, 2)$ type (d_4) , et de ses deux sous-masques antérieur (M_a) et postérieur (M_p) . Les parcours de ces deux masques (M_a) (M_p) , dans chacune des deux passes, sont respectivement de types lexicographique (\Rightarrow) (haut-gauche vers bas-droit) et lexicographique inverse (\Leftarrow) (bas-droit vers haut-gauche). Au cours de ces deux passes, le pixel central (p) est obtenu à partir de l'application des masques de chanfrein antérieur (M_a) et postérieur (M_p) au voisinage (P_n) (équation (1.14)), et des équations respectives (d'_c) (d_c) (équation (1.15)). La distance de chanfrein (d_c) est donc obtenue à l'issue de la deuxième passe. La figure (1.11) (page 11) donne un exemple de résultat de transformée de distance avec un masque de chanfrein $(1, 2)$.

$$M = \begin{array}{|c|c|c|} \hline 2 & 1 & 2 \\ \hline 1 & 0 & 1 \\ \hline 2 & 1 & 2 \\ \hline \end{array} \quad M_a = \begin{array}{|c|c|c|} \hline 2 & 1 & 2 \\ \hline 1 & 0 & \\ \hline \end{array} \quad M_p = \begin{array}{|c|c|c|} \hline & 0 & 1 \\ \hline 2 & 1 & 2 \\ \hline \end{array} \quad (1.13)$$

$$P_n = \begin{array}{|c|c|c|} \hline p_1 & p_2 & p_3 \\ \hline p_0 & p & p_4 \\ \hline p_7 & p_6 & p_5 \\ \hline \end{array} \quad M_a = \begin{array}{|c|c|c|} \hline e_1 & e_2 & e_3 \\ \hline e_0 & 0 & \\ \hline \end{array} \quad M_p = \begin{array}{|c|c|c|} \hline & 0 & e_4 \\ \hline e_7 & e_6 & e_5 \\ \hline \end{array} \quad (1.14)$$

$$d'_c(p, P_n) = \min(\bigcup_{n=0}^3 p_n + e_n) \quad d_c(p, P_n) = \min(\bigcup_{n=4}^7 p_n + e_n) \quad (1.15)$$

Basées sur le résultat de la carte des distances, les méthodes de squelettisation par transformée de distance extraient l'axe médian correspondant au squelette [Klette 03] [Foskey 03] [Ronse 04]. La notion d'axe médian est basée sur l'idée que chaque forme peut être représentée par un ensemble de boules (ou disques) maximales. La définition (2) suivante formalise cette notion [Ronse 04]. La figure (1.13) donne un exemple d'axe médian extrait, et quelques boules maximales représentatives. Comme le montre la figure (1.13), l'axe médian correspond donc à un squelette de la forme où chaque pixel est attribué d'une intensité proportionnelle à l'épaisseur du tracé initial.

²⁶Nous reportons le lecteur à [Ronse 04] pour une justification de cet algorithme.

Définition 2 L'axe médian (A_m) d'une forme (F) est composé de tous les points (c) de (F) qui sont centres d'une boule incluse dans (F), et maximale pour l'inclusion, et dont la distance au bord (r) de (F) est atteinte en au-moins deux points (q_1) et (q_2) tel que $d(c, q_1) = d(c, q_2) = r$.

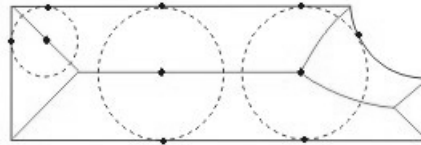


FIG. 1.13 – Axe médian et boules maximales représentatives

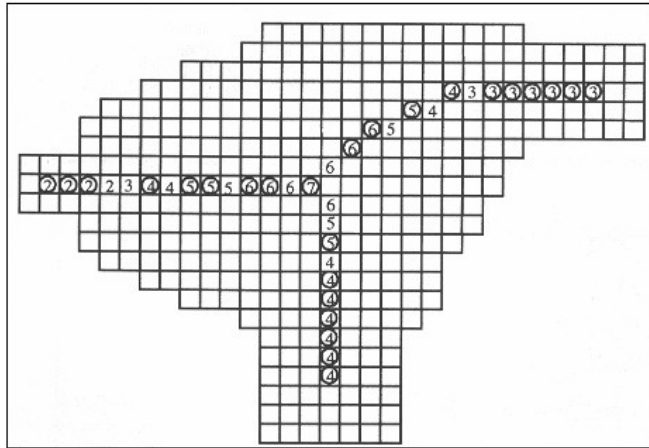


FIG. 1.14 – Axe médian discret

L'extraction de l'axe médian de la carte des distances se fait alors par test de l'intensité (ou distance) de chacun des pixels de la forme dans la carte [Baja 94a]. Ce test est fonction du masque de chanfrein (a, b) utilisé lors du calcul de la carte (figure (1.12) (c)), et du voisinage (P_n) du pixel (p) (équation (1.14)). Il s'agit en effet d'extraire les pixels médian d'intensité maximale répondant à la définition (2). Ceux-ci répondent alors aux contraintes présentées dans l'équation (1.16)²⁷. Cependant, dans le cas d'un espace discret (Z) comme les cartes des distances, la discrétisation provoque une rupture de connexion dans l'axe médian extrait. La figure (1.14) donne un exemple d'axe médian non-connexe extrait [Ablameyko 00], les pixels centres des boules maximales (répondant à l'équation (1.16)) sont représentés par des (\circ). Aussi, durant l'extraction de l'axe médian, différents pixels dits de connexion sont également extraits²⁸. La figure (1.14) présente des pixels de connexion dans le voisinage des pixels centres des boules maximales (\circ) [Ablameyko 00].

²⁷Nous reportons le lecteur à [Baja 94a] pour une justification et présentation de ces aspects.

²⁸Nous reportons le lecteur à [Ablameyko 00] sur ces aspects.

$$\forall p_{i=\{0,2,4,6\}} < (p+a) \quad \wedge \quad \forall p_{i=\{1,3,5,7\}} < (p+b) \quad \Rightarrow \quad p \in A_m \quad (1.16)$$

Les algorithmes de squelettisation par transformée de distance (contrairement à ceux par amincissements successifs) permettent un temps de calcul rapide et conservent l'information sur l'épaisseur [Klette 03]. En contrepartie, ils ne garantissent pas d'obtenir un squelette exact : le squelette peut être non connexe et composé d'arcs et de courbes d'épaisseurs non-unitaires (> 1 pixel). Aussi, le choix d'utilisation entre les algorithmes de squelettisation par amincissement successifs/transformation de distance dépend du compromis précision/rapidité souhaité [Lam 92]. Quel que soit l'algorithme de squelettisation employé, la squelettisation reste très sensible aux bruits [Chen 96]. Ce bruit est de trois natures comme le montre la figure (1.15) : la distorsion des jonctions (a), les barbules (b), et la distorsion des courbes (c). C'est pour cette raison que, malgré le nombre important de travaux réalisés sur la squelettisation ([Lam 92], [Lam 95], [Klette 03], ...), celle-ci reste un domaine de recherche actif [Kégl 02]. En effet, de nouvelles approches sont développées²⁹ comme : la squelettisation par inférence statistique [Zhu 99], la Veinerization [Deseilligny 98], la squelettisation à base de règles [Ahmed 02], la squelettisation par multi-résolution [Borgefors 01], ...

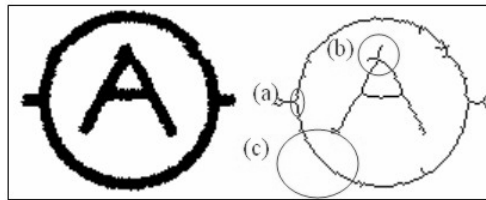


FIG. 1.15 – Bruits typiques issus de la squelettisation
(a) distorsion des jonctions (b) barbules (c) distorsion des courbes

1.3.3 La construction du graphe de squelette

La squelettisation permet d'obtenir des images de représentation du squelette. Afin d'en extraire des primitives graphiques, les méthodes à base de squelettisation emploient une deuxième étape de construction du graphe de squelette³⁰. On distingue trois types d'éléments dans un squelette (figure (1.16) (a)) : les extrémités (pixels 1-connexe), les jonctions (pixels n -connexe avec $n \geq 3$), et les chaînes (ensemble de pixels 2-connexes) reliant les extrémités et/ou les jonctions. L'étape de construction extrait et structure ces différents éléments en un graphe³¹. La figure (1.16) donne un exemple de graphe de squelette construit (b) à partir de l'image (a). Les noeuds représentent les chaînes et les jonctions, et les arcs leurs relations de connexion.

²⁹Ces approches sont simplement mentionnées, nous reportons le lecteur aux références citées.

³⁰building skeleton graph [Popel 02]

³¹Nous reportons le lecteur à l'Annexe A pour une introduction sur les graphes.

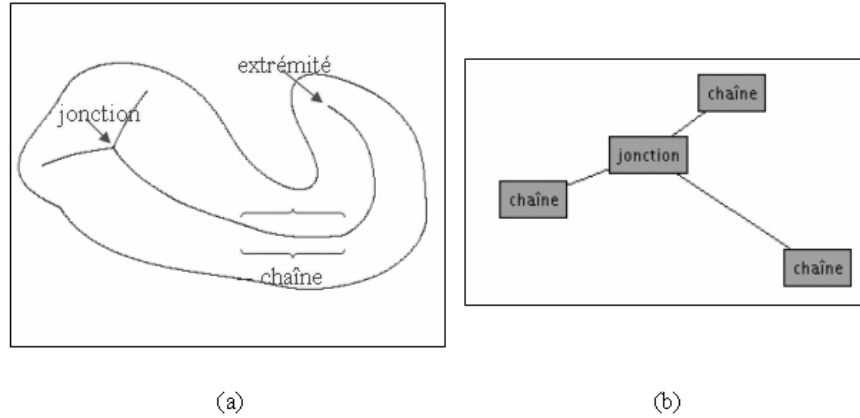


FIG. 1.16 – Graphe de squelette
(a) squelette (b) graphe

La construction du graphe de squelette est proche du chaînage³² des contours. Elle se spécialise à celui-ci en ce qui concerne le traitement des jonctions, et la structuration en graphe des jonctions et des chaînes de pixels³³. Malgré le volume important de travaux sur la squelettisation ([Lam 92], [Zhang 93], [Lam 95], [Cowell 01], [Klette 03], ...) il existe peu de travaux dédiés à l'étape de construction du graphe de squelette [Tombre 00]. Nous en citons ici quelques uns que nous estimons représentatifs : [Heutte 96], [Liu 99], [Tombre 00], [Lau 02], [Lin 02] et [Popel 02]. La problématique principale de la construction du graphe de squelette réside dans la distorsion des jonctions³⁴ produite par l'étape de squelettisation. Cette distorsion a pour conséquence principale la construction de jonctions multiples (accumulation de plusieurs pixels jonctions dans le même voisinage). La figure (1.17) donne différents exemples de jonctions multiples [Lin 02]. Selon notre point de vue deux approches de construction du graphe de squelette se distinguent dans la littérature : par détection des jonctions ([Tombre 00] [Liu 99] [Lin 02] [Popel 02]), et par suppression/reconstruction des jonctions [Lau 02]. Nous présentons chacune d'entre elles dans la suite de cette section.

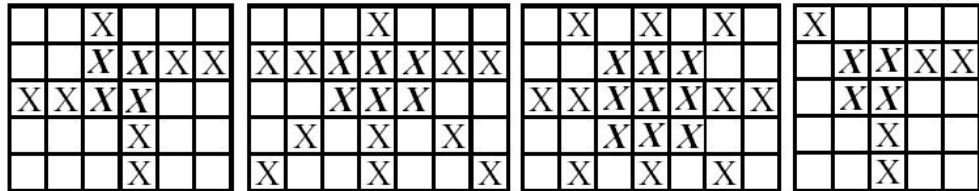


FIG. 1.17 – Jonctions multiples

³²Nous reportons le lecteur page 5 sur ces aspects.

³³Nous ne présentons ici que ces aspects, et reportons le lecteur à [Nunes 00] sur le chaînage.

³⁴De fausses jonctions peuvent également apparaître (épaisseurs > 1 pixel) [Lassaulzais 97].

La construction du graphe de squelette par détection des jonctions est l'approche la plus communément utilisée ([Heutte 96] [Liu 99] [Tombre 00] [Lin 02] [Popel 02]). Elle se base sur l'analyse du voisinage des pixels pour la détection des pixels jonction, différentes méthodes sont utilisées :

La méthode directe [Popel 02] consiste à construire le graphe exact du squelette. Ce graphe est qualifié de "exact" car il regroupe l'intégralité des pixels du squelette. [Popel 02] calcule pour cela le nombre de voisins connexes (N_n) dans le voisinage (P_n) de chacun des pixels du squelette (équation (1.17)). Il est ainsi possible d'identifier les pixels extrémités ($N_n = 1$), chaînes ($N_n = 2$), et jonctions ($N_n \geq 3$). Ces pixels sont par la suite structurés par analyse de leur voisinage (P_n) et selon leur classe.

$$P_n = \begin{array}{|c|c|c|} \hline p_3 & p_2 & p_1 \\ \hline p_4 & p & p_{0,8} \\ \hline p_5 & p_6 & p_7 \\ \hline \end{array} \quad N_n = \sum_{i=0}^7 p_i \quad N_c = \frac{1}{2} \times \sum_{i=0}^7 |p_{i+1} - p_i| \quad (1.17)$$

D'autres méthodes [Heutte 96] [Tombre 00] appliquent des règles de parcours afin de superviser la construction du graphe. [Tombre 00] parcourt (et chaîne) prioritairement les pixels voisins 4-connexes, puis les 8-connexes (non connexes à un voisin 4-connexes). Les pixels chaînés sont itérativement effacés. De manière similaire, [Heutte 96] utilise différents éléments structurants afin d'étiqueter les pixels jonctions. L'équation (1.18) présente les éléments structurants utilisés pour la détection de la jonction de type (Y) dans 4 orientations. Les pixels s'appariant avec les éléments structurants correspondent alors aux pixels jonctions. Les voisinages décrits par les éléments structurants sont utilisés pour déterminer les directions suivantes de chaînage du squelette.

$$\begin{array}{|c|c|c|} \hline 1 & * & 1 \\ \hline * & 1 & * \\ \hline * & 1 & * \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline * & 1 & * \\ \hline * & 1 & * \\ \hline 1 & * & 1 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline 1 & * & * \\ \hline * & 1 & 1 \\ \hline 1 & * & * \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline * & * & 1 \\ \hline 1 & 1 & * \\ \hline * & * & 1 \\ \hline \end{array} \quad (1.18)$$

Une dernière approche [Liu 99] [Lin 02] consiste à calculer le nombre de croisements³⁵. Ce nombre de croisement (N_c) a été introduit par [Rutovitz 66], l'équation (1.17) en précise la définition³⁶. Partant de cette définition trois cas de figure peuvent se produire selon que le pixel soit : un pixel extrémité ($N_c = 1$), un pixel chaîne ($N_c = 2$), un pixel jonction comportant N_c croisements ($N_c \geq 3$). Le calcul du nombre de croisements permet ainsi de détecter les pixels redondants dans les jonctions multiples et de les ignorer au cours du chaînage.

³⁵ crossing number

³⁶ Une autre définition existe [Hilditch 69].

Les jonctions sont dans certains cas de structures complexes, leur détection peut donc parfois s'avérer difficile [Liu 99]. Une autre approche utilisée consiste à procéder par suppression/reconstruction des jonctions [Lau 02]. Elle procède alors en deux étapes : une étape de suppression des pixels jonctions suivie d'une étape de reconstruction des jonctions. La figure (1.18) illustre le principe de l'étape de suppression. [Lau 02] recherche les pixels jonctions sur l'image (a) par calcul du nombre de croisement (N_c) (équation (1.17)). Ceux-ci sont par la suite supprimés du squelette (b), celui-ci est alors composé uniquement de pixels 1-connexes (les pixels extrémités) et 2-connexes (les pixels chaînes). Les pixels extrémités (e_i) sont par la suite utilisés durant l'étape de reconstruction des jonctions (équation (1.19)). Pour ce faire, [Lau 02] calcule la distance euclidienne (d_e) entre toutes les extrémités (e_i). Le regroupement de deux extrémités différentes est décidé par application du seuil ($2 \times W_m$) (équation (1.19)). Dans ce seuil, (W_m) correspond à la moyenne des degrés d'intériorité obtenue par analyse du squelette par transformée de distance [Lau 02]. Durant cette étape de reconstruction, des ruptures de connexions aux jonctions peuvent être corrigées. La figure (1.18) en donne un exemple : la chaîne 5 (b) initialement non-connectée à la jonction (a) peut être regroupée en raison de sa proximité aux autres extrémités.

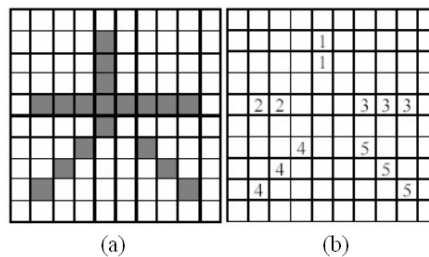


FIG. 1.18 – (a) image (b) suppression et chaînage

$$e_1(x_1, y_1) \quad e_2(x_2, y_2) \quad d_e = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad d_e \leq 2 \times W_m \quad (1.19)$$

Vu l'absence de travaux de synthèse sur la construction du graphe de squelette [Tombre 00] il est difficile de dresser une évaluation des approches. Selon notre point de vue, l'approche par suppression/reconstruction des jonctions [Lau 02] semble plus adaptée que celle par détection lorsque le squelette comporte des jonctions multiples complexes (figure (1.17)). En effet, celles-ci forment des amas de pixels difficile à détecter sans recours à une approche plus globale. En contrepartie, cette méthode peut entraîner différents problèmes de connexité : des ruptures (séparation chaînes/jonctions), des fausses connexions (mauvais groupement chaînes/jonctions). Le choix entre les approches par détection et par suppression/reconstruction sera donc fonction, de la nature des jonctions à traiter, et du compromis précision/filtrage souhaité. Dans les deux cas, les problèmes de distorsion des jonctions du squelette rendent de toute façon la construction du graphe complexe et ambigu [Lau 02].

1.4 Les méthodes à base de parcours de formes

1.4.1 Introduction

Les méthodes à base de parcours de formes³⁷ constituent une alternative aux méthodes à base de squelettisation [Dori 99]. Elles permettent l'extraction de graphes représentant les axes médians des composantes en une étape. Les premiers travaux sur ces méthodes datent de la fin des années 1980 [Baruch 88]³⁸. Ils se sont développées depuis lors, on en compte aujourd'hui plusieurs significatifs : [Mullot 91], [Paquet 92], [Lefrere 93], [Nieuwenhuizen 94], [Ogier 94], [Chiang 98], [Dori 99], [Yoon 01], et [Song 02b]. Certains états de l'art y sont également en partie consacrés [Wenyin 99] [Song 03]. Malgré ce développement, ces méthodes restent largement moins utilisées que celles à base de squelettisation. Une majorité d'entre elles ([Mullot 91] [Paquet 92] [Nieuwenhuizen 94] [Dori 99] [Song 02b]) permettent également l'approximation mathématique des axes médians extraits en vecteurs et/ou arcs. Aussi, elles sont également qualifiées dans la littérature de méthodes de vectorisation directe³⁹ [Ablameyko 00]. La figure (1.19) donne un exemple de représentation vectorielle d'une image par la méthode de parcours de formes de [Dori 99].

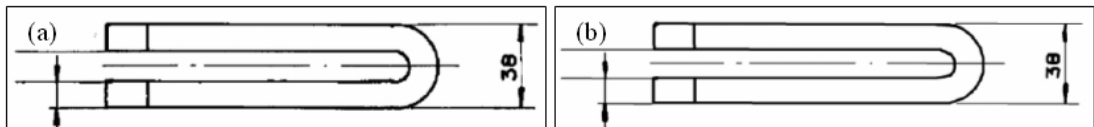


FIG. 1.19 – Parcours de [Dori 99] (a) image (b) représentation vectorielle

Selon notre point de vue, les méthodes de parcours de formes peuvent être décomposées en deux parties : le suivi de traits et la détection des jonctions. Le tableau (1.1) propose une taxinomie des travaux selon cette décomposition. Les méthodes employées dans chacune des parties sont basées sur l'utilisation d'éléments de parcours. Ceux-ci peuvent être (figure (1.20)) de type pixellaire (a) [Dori 99] et/ou surfacique (b) [Chiang 98]. Nous présentons les différentes méthodes de suivi de traits et de détection de jonctions selon ces éléments de parcours dans la suite de cette section.

	Pixellaire	Surfacique
Suivi de traits	[Paquet 92] [Nieuwenhuizen 94] [Dori 99] [Song 02b]	[Mullot 91] [Chiang 98] [Yoon 01]
Détection de jonctions	[Lefrere 93] [Dori 99] [Song 02b]	[Mullot 91] [Nieuwenhuizen 94] [Ogier 94] [Chiang 98] [Yoon 01]

TAB. 1.1 – Taxinomie des travaux de parcours de formes

³⁷ shape tracking

³⁸ Méthode à base parcours par fenêtre proche des méthodes de parcours de formes [Ogier 94].

³⁹ direct vectorisation

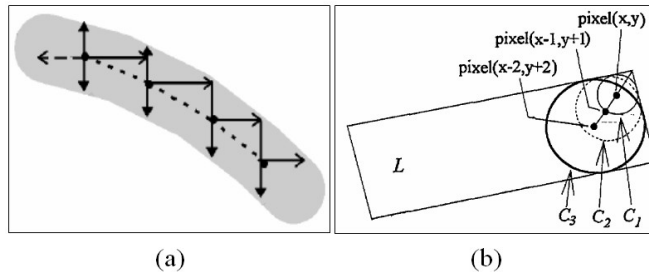


FIG. 1.20 – Éléments de parcours pixellaire/surfacique
(a) pixellaire (b) surfacique

1.4.2 Suivi de traits

Le suivi de traits est la partie centrale des méthodes de parcours de formes. Il existe différents travaux sur le suivi de traits (tableau (1.1)). Ils emploient cependant une suite d'étapes (E_i) identiques résumées dans la liste qui suit. La principale différence entre ces travaux réside dans l'approche utilisée pour l'étape (E_3)⁴⁰. Cette étape (E_3) est centrale au suivi de traits. Elle peut être de deux natures (tableau (1.1)) : pixellaire et surfacique.

E_1 Localisation d'un trait

E_2 Recherche d'un point d'entrée dans le trait

E_3 Suivi du trait par pas successif jusqu'à la rencontre d'une fin de trait ou d'une jonction.

Les suivis pixellaires emploient des éléments de parcours de type pixel afin d'explorer les traits. Ces éléments pixellaires sont alors pilotés durant le suivi en fonction du type de modèle de trait recherché : vectoriel ou circulaire [Song 02b] :

Le suivi vectoriel emploie deux familles de méthodes (figure (1.21)) : par analyse des contours (a) [Paquet 92] [Nieuwenhuizen 94] [Song 02b] et par déplacements orthogonaux (b) [Dori 99]. Les méthodes de suivi par analyse des contours [Paquet 92] [Nieuwenhuizen 94] [Song 02b] sont très similaires. Nous présentons ici la méthode initiale développée par [Paquet 92]. Celui-ci progresse de façon médiane dans le trait (figure 1.21 (a)). Le vecteur choisi est celui permettant le plus grand déplacement au sens des directions de [Freeman 61]. Ce déplacement est proportionnel à l'épaisseur du trait. Il est possible de régler la précision du déplacement par pondération. À chaque fin de déplacement, le point de départ du déplacement suivant est recentré. Ce recentrage se calcule par tracé de la perpendiculaire à la direction de [Freeman 61] précédemment empruntée, et de l'analyse de son intersection avec les contours du trait. [Dori 99] dans son

⁴⁰Nous abordons ici que cette étape et reportons le lecteur aux articles cités pour (E_1) et (E_2).

algorithme de SPV⁴¹ suit le trait à l'aide de deux tracés orthogonaux (figure 1.21 (b)). Le premier des tracés est utilisé pour l'extraction des traits verticaux, et l'autre pour l'extraction des traits horizontaux. Les informations sur les deux tracés sont fusionnées en cours de suivi. Cet algorithme se nommait auparavant *orthogonal zigzag* [Chai 92]⁴². Ce nom exprime clairement le type de suivi réalisé sur le trait par cet algorithme.

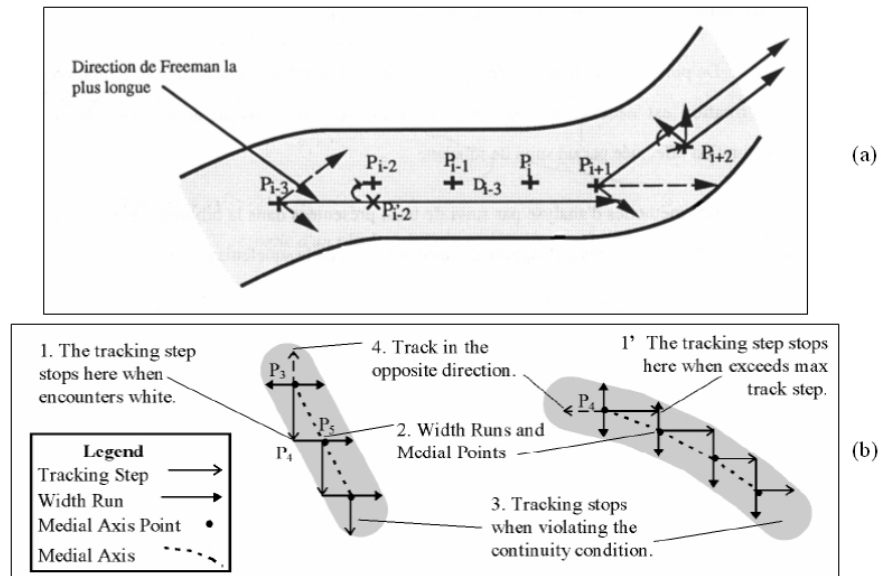


FIG. 1.21 – Suivi de trait vectoriel
(a) par analyse des contours (b) par déplacements orthogonaux

Il existe peu de travaux sur le suivi circulaire, nous en avons référencé qu'un seul celui de [Song 02b]. Celui-ci emploie un algorithme en trois étapes : détection d'arc potentiel, suivi par ajustement dynamique du centre, et calcul du centre suivant. L'étape de suivi est la partie centrale de cet algorithme⁴³. Celle-ci est amorcée (figure (1.22)) suite à la détection d'un arc potentiel $\{P_1, P_2, O\}$. Le trait est alors suivi en fonction d'un tracé théorique (*circular tracking path*). Contrairement aux méthodes de suivi précédentes, le suivi est donc ici continu (il ne procède pas par pas). Le tracé théorique est calculé par l'algorithme de [Bresenham 77]. L'axe médian (*median axis*) du trait est également extrait durant le suivi par recherche des intersections entre le rayon et les contours du trait. La distance entre le tracé théorique et l'axe médian est constamment re-calculée. Une distance trop importante correspond alors à un écart trop fort entre le tracé théorique et l'axe médian. Dans ce cas le centre (O) du cercle est re-calculé : [Song 02b] parle alors d'ajustement dynamique.

⁴¹ Sparse Pixel Vectorisation

⁴² Nous reportons le lecteur à [Chai 92] et [Dori 99] sur l'évolution de cet algorithme.

⁴³ Nous la présentons ici et reportons le lecteur à [Song 02b] sur les deux autres étapes.

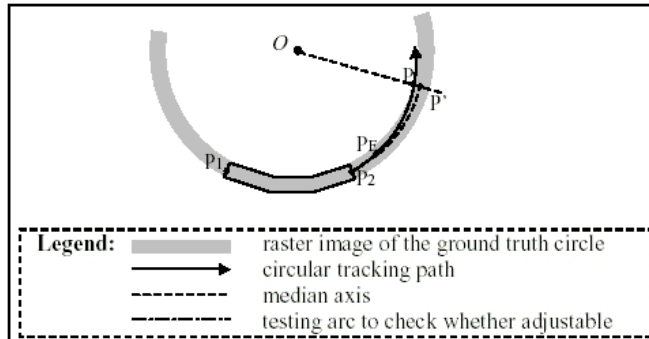


FIG. 1.22 – Suivi de trait circulaire

Le suivi surfacique [Mullot 91] [Chiang 98] [Yoon 01] exploite des éléments de parcours basés sur l'analyse de la surface afin d'explorer le trait. L'élément de parcours utilisé peut être de différentes natures : [Mullot 91] et [Chiang 98] emploient un cercle, [Yoon 01] un élément gaussien. De par la nature surfacique des éléments de parcours, certaines méthodes [Mullot 91] et [Chiang 98] marquent également les pixels des traits durant les suivis. Nous présentons ici la méthode de [Mullot 91] qui nous semble la plus représentative. Une fois un point d'entrée trouvé dans le trait, celui-ci exploite un cercle afin de suivre le trait. Il recherche pour cela (figure (1.23)) le cercle inscrit maximum partant de ce point. Le cercle suivant est obtenu par calcul des cercles voisins dans les 8 directions de [Freeman 61]. Le cercle voisin de densité maximum est retenu. Les cercles calculés ne correspondent cependant pas toujours à des cercles inscrits (a). De même, dans le cas où ils sont inscrits, ils ne correspondent pas toujours aux cercles inscrits de surface maximale commune. [Mullot 91] procède alors à une étape de correction de façon à re-centrer le cercle (b). Il trace la perpendiculaire à la direction de [Freeman 61] retenue précédemment, et recherche le point médian du trait sur cette perpendiculaire. Les méthodes de [Chiang 98] et [Yoon 01] sont proches de celle de [Mullot 91]. Les différences résident dans le mode de progression dans le trait (continu [Chiang 98]) et dans l'élément de parcours employé (gaussien [Yoon 01]).

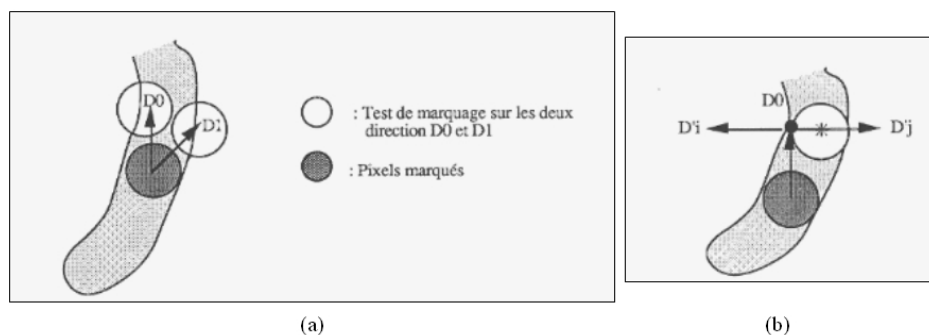


FIG. 1.23 – Suivi de trait surfacique

Il est difficile d'effectuer une évaluation des méthodes de suivi de traits vu le manque de travaux comparatifs [Song 03]. Selon notre point de vue, les méthodes de suivi vectoriel donnent une meilleure approximation de l'axe médian que les méthodes de suivi surfacique. En contrepartie, les méthodes de suivi surfacique sont moins sensibles aux bruits (irrégularité des contours, bruit impulsif, ...). Le choix entre les deux résidera donc du compromis précision/robustesse souhaité. L'autre différence réside dans le pas de progression adopté par la méthode : continu [Chiang 98], régulier [Mullot 91] [Yoon 01], variable en favorisant le plus grand déplacement [Paquet 92] [Nieuwenhuizen 94] [Dori 99] [Song 02b]. Cependant un pas de progression important ne garantit pas une approximation suffisante de l'axe médian extrait. En effet plusieurs de ces méthodes [Paquet 92] [Dori 99] exploitent une seconde étape de polygonalisation à l'issue du suivi afin d'affiner cette approximation. Dans tous les cas les méthodes de suivi de traits permettent une bonne approximation des axes médians et sont de faible complexité algorithmique [Wenyin 99] [Song 03]. Cette approximation dépend en contrepartie de la bonne détection du début et fin de trait ainsi que des jonctions.

1.4.3 Détection de jonctions

La détection de jonctions est complémentaire au suivi de traits [Ogier 94]. Elle s'exécute conjointement à celui-ci afin de détecter les jonctions durant les suivis, ainsi que les germes de trait aux voisinages des jonctions. Ces germes de trait correspondent à des débuts de traits qui n'ont pas encore été suivis. La méthode de parcours supervise alors le suivi de traits et la détection de jonctions afin de déterminer de l'arrêt et de la reprise des suivis. Parallèlement, elle construit le graphe de la forme parcourue. Ce graphe structure les résultats de la détection de jonctions du suivi de traits. La détection de jonctions emploie les mêmes éléments de parcours que le suivi de traits (tableau (1.1)) : vectoriel ou surfacique.

Les méthodes de détection vectorielle [Lefrere 93] [Dori 99] [Song 02b] utilisent des projections vectorielles afin de détecter les jonctions. Ces projections sont effectuées selon les directions de [Freeman 61]. La détection d'une jonction est basée sur un critère de sur-épaisseur du trait suivi. Cette sur-épaisseur correspond à des sur-longueurs des vecteurs projetés. Nous présentons ici les méthodes de [Lefrere 93] et [Dori 99]⁴⁴.

[Lefrere 93] détecte les jonctions par recherche de la sur-épaisseur dans 6 directions de [Freeman 61] : 4 sur les diagonales et 2 sur la verticale (figure (1.24)). Dès qu'il détecte une épaisseur supérieure à l'épaisseur du trait en cours de suivi (multiplié par $\sqrt{2}$ pour les diagonales), il lance une procédure de recherche des jonctions. Il explore l'axe de la sur-épaisseur détectée de façon à vérifier s'il s'agit d'un germe de trait. Cette exploration vérifie l'existence des bords du nouveau trait ainsi que les possibilités de continuation dans celui-ci. La position du germe de trait détecté est sauvegardée pour une exploration future, et l'exploration du trait en cours se poursuit.

⁴⁴La méthode de [Song 02b] est dérivée de celle de [Dori 99].

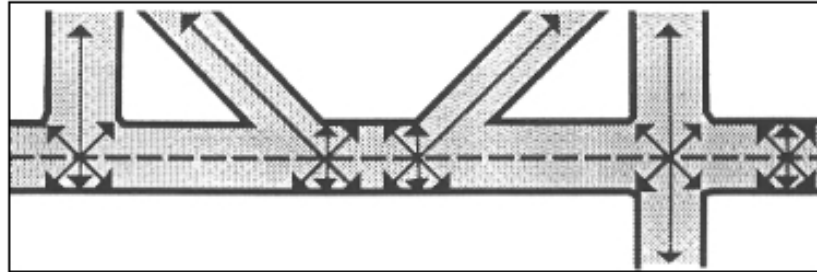


FIG. 1.24 – Détection vectorielle de [Lefrere 93]

[Dori 99] teste différentes conditions $\{C_1, C_2, C_3, C_4\}$ susceptibles de stopper son suivi. La rencontre d'une jonction durant le suivi viole en effet une ou plusieurs de ces conditions :

- C_1 Variations de l'épaisseur dans les deux directions verticales.
- C_2 Croisement d'un suivi déjà réalisé.
- C_3 Variation du pas de progression dans le trait : considérant la nature en zigzag du suivi une variation d'épaisseur entraînera une modification du pas de progression.
- C_4 La longueur du pas de progression doit être supérieure à zéro.

Les trois premières conditions $\{C_1, C_2, C_3\}$ sont propres à la détection de jonctions. La quatrième (C_4) est exploitée pour la détection d'une fin de trait. Lorsqu'une jonction est détectée, [Dori 99] parcourt celle-ci par tâtonnement (figure 1.25). Ce parcours s'effectue sur la portion de l'axe médian correspondant à la jonction et à son voisinage. À partir d'un point appartenant à la jonction (a), [Dori 99] parcourt alors celle-ci vers l'amont (b) afin de retrouver le dernier point stable du trait (c). Il parcourt ensuite la jonction vers l'aval (d) afin de trouver le germe de trait suivant.

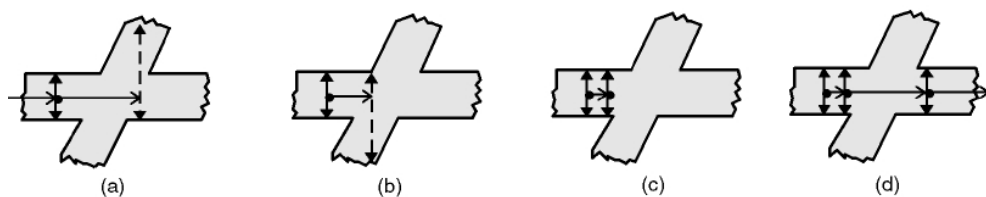


FIG. 1.25 – Détection vectorielle de [Dori 99]

Les méthodes de détection surfacique [Mullot 91] [Nieuwenhuizen 94] [Ogier 94] [Chiang 98] [Yoon 01] exploitent des zones de surface pour détecter et parcourir les jonctions. On distingue deux catégories de méthodes selon qu'elles recherchent : la connexité [Mullot 91] [Chiang 98] [Yoon 01] ou la rupture de connexité [Nieuwenhuizen 94] [Ogier 94].

Celles recherchant la connexité [Mullot 91] [Chiang 98] [Yoon 01] sont proches des méthodes de détection vectorielle précédemment présentées⁴⁵. À chaque étape du suivi, le 8 voisinage de la zone de surface parcourue (cercle ou élément gaussien) est analysé. Une jonction est détectée si le suivi peut se poursuivre dans au moins deux directions (soit une connexité supérieure à deux). [Mullot 91] par exemple⁴⁶ autorise durant le suivi la détection de plusieurs cercles inscrits maximum (figure (1.26)). La connexité des cercles est analysée durant le suivi, les cercles 3-connexes et plus correspondent alors à des cercle jonctions.

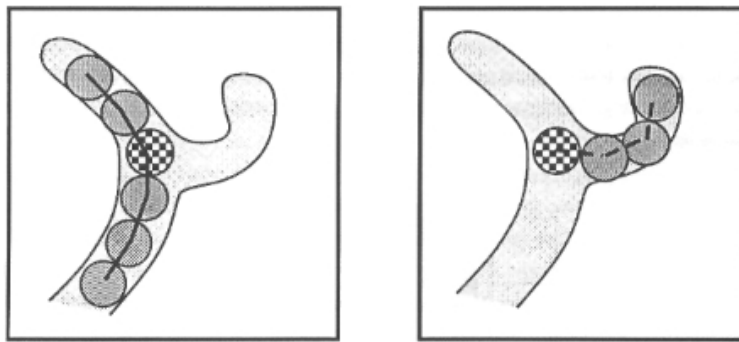


FIG. 1.26 – Détection surfacique de [Mullot 91]

Les méthodes de [Ogier 94] et [Nieuwenhuizen 94] exploitent la rupture de connexité pour la détection des jonctions. Nous présentons ici la méthode de [Ogier 94]⁴⁷. Celui-ci soulève le problème posé par les méthodes de détection de jonctions (vectorielle et surfacique) précédemment présentées (figure (1.27)). En effet, les cas de sur-épaisseur des traits (a) tendent à provoquer de fausses détections de jonctions avec ces méthodes. Afin de résoudre ce problème [Ogier 94] exploite, parallèlement à son suivi de trait, un suivi surfacique appliqué aux bords du trait (b). Ce suivi emploie des cercles de diamètre identique (fixé à 4 pixels dans son système), et espacés à intervalle régulier de un rayon. Une jonction est détectée lorsqu'il y a perte de connexité dans le suivi des bords (c). En effet, dans le cas d'une jonction, la progression sur les bords du trait ne suit plus la progression du centre du trait. [Ogier 94] exploite alors l'absence de connexité comme critère de détection d'une jonction. Il recherche ensuite les germes de trait de la jonction en traçant un cercle circonscrit à cette jonction (d). Il localise les germes de trait sortants, en recherchant les intersections de ces germes avec le cercle⁴⁸ circonscrit de la jonction. Cette méthode permet ainsi une meilleure approximation des jonctions que les méthodes de squelettisation car elle repose sur l'utilisation d'information plus globale [Nieuwenhuizen 94] [Ogier 94].

⁴⁵Nous les abordons sommairement ici et reportons le lecteur à [Mullot 91] [Chiang 98] [Yoon 01].

⁴⁶[Chiang 98] et [Yoon 01] utilisent des méthodes proches de [Mullot 91].

⁴⁷La méthode de [Nieuwenhuizen 94] est proche de celle de [Ogier 94].

⁴⁸[Nieuwenhuizen 94] en exploite plusieurs à la manière de sondes circulaires.

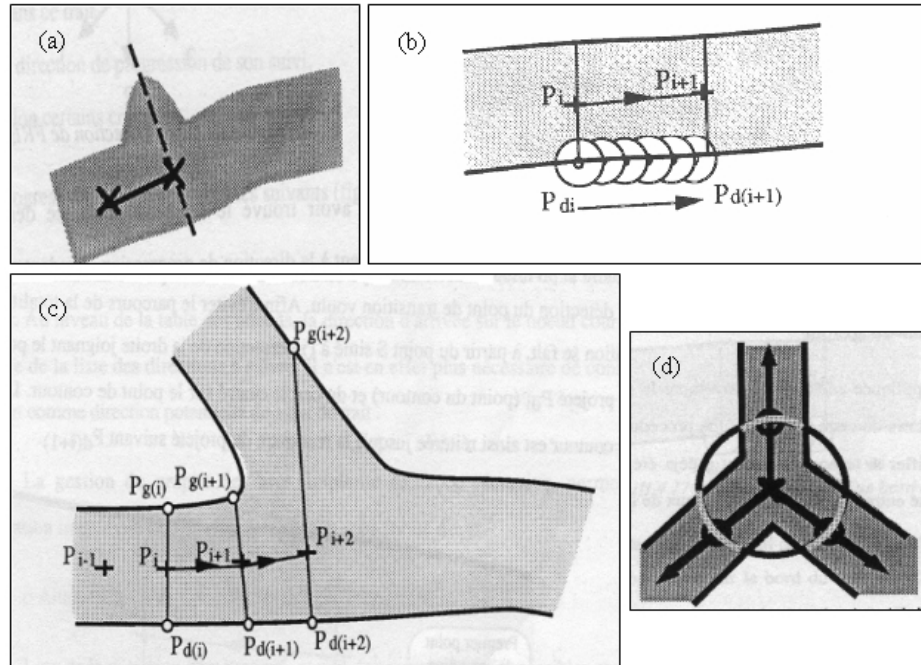


FIG. 1.27 – Détection surfacique de [Ogier 94]

1.5 Les méthodes à base de décomposition en plages

Les méthodes à base de décomposition en plages⁴⁹ ont été introduites par [Agrawala 77]. Les premiers travaux portant spécifiquement sur ces méthodes ont été proposés par [Pavlidis 78]. Depuis lors différents travaux y sont dédiés⁵⁰ : [Zenzo 89], [Boatto 92], [Monagan 93], [Lin 95], [Zenzo 96], [Burge 98], [Fan 98], [Zhong 99], [Lanza 01], et [Xue 01]. [Wenyin 99] y consacre également une partie de son état de l'art. Celui-ci précise que malgré ce développement, ces méthodes restent largement moins utilisées que celles à base de squelettisation.

Parmi les travaux sur ces méthodes différentes formalisations proches ont été proposées : [Monagan 93], [Zenzo 96], [Burge 98], [Fan 98], ... Nous présentons ici celle de [Burge 98] qui nous semble la plus représentative. Celui-ci propose les définitions⁵¹ (3) et (4) de la plage et de la relation d'adjacence entre plages. S'appuyant sur ces deux définitions, il présente alors la définition (5) de la classification des plages. La figure (1.28) illustre ces différentes définitions avec la décomposition en plages verticales d'un extrait de plan de cadastre [Boatto 92].

⁴⁹ runs

⁵⁰ Cette liste est non exhaustive.

⁵¹ Nous reportons le lecteur à [Burge 98] pour les versions anglaises de ces définitions.

Définition 3 Une plage est une séquence maximale de pixels noirs dans une direction horizontale ou verticale. Une plage est définie selon quadruplet $\{d, x, y, l\}$ avec (d) la direction, (x, y) le point d'origine de la plage, et (l) sa longueur.

Définition 4 Deux plages $p = \{d_p, x_p, y_p, l_p\}$ et $q = \{d_q, x_q, y_q, l_q\}$ sont adjacentes si :

- $d_p = d_q$
- $(|y_p - y_q| = 1) \wedge (|x_p + l_p - x_q - l_q| \geq 1)$ pour des plages horizontales
- $(|x_p - x_q| = 1) \wedge (|y_p + l_p - y_q - l_q| \geq 1)$ pour des plages verticales

Définition 5 Selon leur relation d'adjacence les plages appartiennent à trois classes :

- La plage extrémité est adjacente à une seule autre plage.
- La plage ligne est adjacente à une seule autre plage sur chacun de ses côtés.
- La plage jonction est adjacente à plus d'une autre plage sur un de ses côtés.

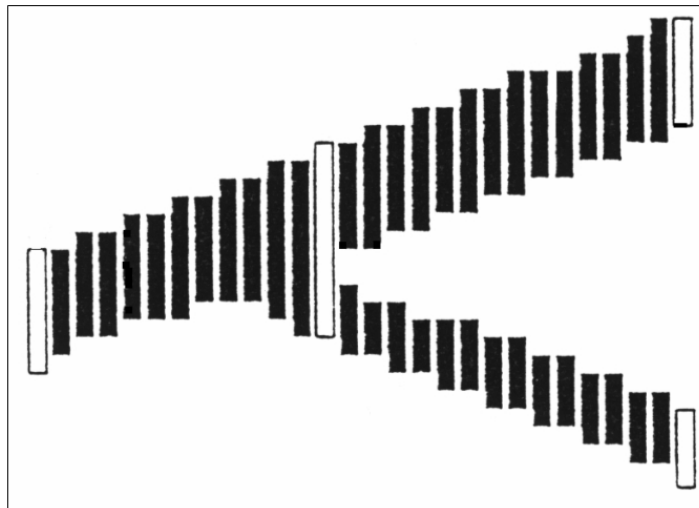


FIG. 1.28 – Décomposition en plages verticales⁵²

La décomposition en plages de l'image se fait par un algorithme de parcours des lignes⁵³ [Rosenfeld 82]. Cet algorithme parcourt (verticalement ou horizontalement) chacune des lignes de l'image afin de détecter les séquences maximales de pixels noirs correspondant aux plages (définition (3)). Lorsqu'une plage est encodée, ses relations d'adjacence (définition (4)) avec les plages de la ligne précédente sont recherchées [Shima 90]. À partir de ces relations, la classification de cette plage est construite (définition (5)). Ces relations d'adjacence sont bijectives, leurs constructions pour les plages d'une ligne (j) sont répercutées pour les plages de la ligne $(j - 1)$. La classification définitive des plages de la ligne $(j - 1)$ n'est donc connue qu'à l'issue du parcours de la ligne (j) .

⁵² Les plages lignes sont représentées en noir, les plages extrémités et jonctions en blanc.

⁵³ scan-line

À partir de la décomposition en plages d'une image, de leurs relations d'adjacence et de leur classification, il est possible de construire différents types de graphes [Pavlidis 78] [Boatto 92] [Burge 98]. [Burge 98] présentent ces différents graphes⁵⁴ dans la figure (1.29). Le LAG⁵⁵ est le graphe défini par [Pavlidis 78]. Celui-ci construit à partir de la décomposition en plages de l'image un graphe d'adjacence de plages. Dans ce graphe, les noeuds représentent les plages et les arcs les relations d'adjacence entre plages. [Boatto 92] a re-défini le LAG en VSG⁵⁶. Dans le VSG [Boatto 92] distingue les plages extrémités, lignes, et jonctions. Les plages extrémités et jonctions sont représentées par des noeuds propres. Les plages lignes ont elles été concaténées et représentées sous forme d'arcs. De cette façon, le VSG se focalise sur la représentation des jonctions, des extrémités, et de leurs relations de connexion. [Monagan 93] et [Burge 98] ont redéfini le VSG en MRG⁵⁷. Ils fusionnent les plages verticales et horizontales afin de construire le graphe. Cette construction hybride est ainsi moins sensible aux écarts d'orientation entre les formes graphiques des documents et les plages [Monagan 93] [Burge 98].

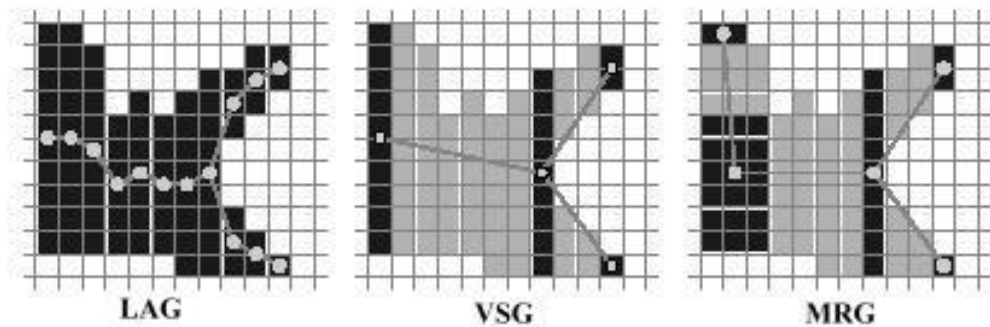


FIG. 1.29 – Graphe de plages

Les constructions du LAG et du VSG reposent sur la seule utilisation des plages, de leurs relations d'adjacence et de leur classification [Pavlidis 78] [Boatto 92]. La construction du MRG met en oeuvre un ensemble de règles complémentaires pour la fusion des plages verticales et horizontales [Monagan 93] [Burge 98]. Ces règles sont présentées dans les définitions⁵⁸ (6), (7), (8), et (9) suivantes. L'application de ces règles [Monagan 93] [Burge 98] sur les plages verticales et horizontales permet d'extraire un graphe composé de région jonction et arc. Chacune des régions est alors composée d'un ensemble de plages. Ces MRG sont inexacts, [Monagan 93] et [Burge 98] emploient par la suite un ensemble de post-traitements⁵⁹ afin de les corriger.

⁵⁴Nous reportons le lecteur à l'Annexe A pour une introduction sur les graphes.

⁵⁵Line Adjacency Graph

⁵⁶Vertical Simple Graph

⁵⁷Mixed Run Graph

⁵⁸Nous omettons ici la règle concernant les connexions entre extrémités des plages [Burge 98].

⁵⁹Nous reportons le lecteur à [Burge 98] sur ces aspects.

Définition 6 Deux plages p et q sont respectivement prédécesseur et successeur si :

- $y_q = y_p + 1$ pour les plages horizontales
- $x_q = x_p + 1$ pour les plages verticales

Définition 7 Selon les relations d'antériorité une plage est :

- régulière si elle a un successeur et un prédécesseur.
- irrégulière si elle n'est pas régulière.

Définition 8 Des plages verticales et horizontales sont :

- conjuguées si elles ont exactement un pixel en commun.
- courtes si elles sont régulières et plus courtes que toutes leurs plages conjuguées.

Définition 9 À partir des définitions (3) à (8) précédentes on définit une région :

- arc comme une séquence maximale de plages adjacentes verticales ou horizontales courtes.
- jonction comme une séquence maximale de plages adjacentes horizontales et de sous-plages qui n'appartiennent pas aux plages verticales et horizontales courtes.

À partir d'un graphe exploitant les plages verticales et horizontales il est possible d'extraire les contours et les squelettes des composantes connexes [Lin 95] [Zhong 99]. L'extraction des contours à partir des plages a été présentée à la page 7 de ce chapitre. La squelettisation se décompose elle en deux étapes⁶⁰, celle des régions arcs et jonctions [Lin 95] [Zhong 99]. La squelettisation des régions arcs extrait, en fonction de leur orientation dominante, les axes médians à partir des plages verticales ou horizontales. Le prolongement de ces axes médians aux régions jonctions détectées (lors de la construction du graphe de plages) permet ensuite de connecter le squelette. La figure (1.30) (a) donne un exemple de caractère et deux résultats de squelettisation par algorithme séquentiel puis par décomposition en plages [Zhong 99]. Cette dernière permet une meilleure squelettisation des régions jonctions en raison de l'utilisation des graphes de plages. Elle utilise ainsi des informations plus globales aux formes [Lin 95] [Zhong 99].

Les représentations à base de plages véhiculent de nombreuses informations pour décrire les formes graphiques : structures (jonctions et extrémités), épaisseurs, régions, contours, squelettes ... Ces descriptions sont ainsi plus complètes que celles obtenues par les méthodes présentées précédemment (squelettisation, détection de contours, suivi de traits). Elles sont utilisées pour différents niveaux d'extraction dans les systèmes : vectorisation et reconnaissance de symboles [Boatto 92] [Monagan 93] [Burge 98] [Lanza 01], reconnaissance de caractères [Lin 95] [Fan 98] [Zhong 99] [Xue 01], mais aussi pour la segmentation [Boatto 92]. La figure (1.30) (b) donne un exemple de segmentation d'un symbole par une méthode à base de décomposition en plages [Boatto 92].

⁶⁰Nous la présentons les brièvement ici et reportons le lecteur à [Lin 95] et [Zhong 99].

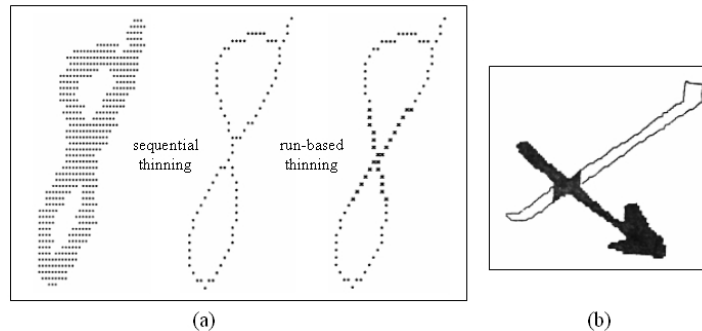


FIG. 1.30 – (a) comparaison squelettisation (b) segmentation

Il n'existe pas de travaux comparatifs sur les méthodes à base de décomposition en plages [Wenyin 99], il est donc difficile d'en effectuer une évaluation. Selon notre point de vue ces méthodes divergent surtout en ce qui concerne la construction des graphes de plages. Le parcours de l'image pour l'encodage des plages présente peu de différences entre les différents travaux [Shima 90]. Concernant la construction des graphes, celles hybrides basées sur les plages verticales et horizontales semblent être privilégiées dans les travaux les plus récents [Burge 98] [Fan 98] [Zhong 99]. Ces dernières sont en effet moins sensibles aux variations d'orientation des formes graphiques composant les documents. Elles permettent de plus d'extraire les squelettes de ces formes. Elles diffèrent cependant entre les différents travaux en ce qui concerne la correction des graphes de plages [Burge 98] et la construction des squelettes [Zhong 99]. Dans tous les cas les méthodes à base de décomposition en plages permettent une description riche des formes graphiques. Elles sont également de complexité algorithmique réduite, l'encodage et la structuration des plages se fait en une ou deux passes sur l'image [Shima 90]. La construction hybride des graphes et l'extraction du squelette nécessitent cependant plusieurs post-traitement des plages [Burge 98] [Zhong 99]. Globalement les méthodes à base de décomposition en plages sont peu adaptées aux multi-orientations des formes graphiques composant les documents. Celles-ci provoquent des ruptures des régions arcs et l'apparition de fausses jonctions dans les graphes de plages extraits [Burge 98].

1.6 Les méthodes à base de segmentation en régions

Les méthodes à base de segmentation en régions segmentent les composantes connexes en un ensemble de régions distinctes. Cette segmentation utilise des méthodes mathématiques afin de labelliser les pixels d'une composante selon leur région d'appartenance. Différentes régions particulières sont considérées : jonctions, lignes, ... [Cao 00] illustre cette segmentation à travers la figure (1.31). Sur cet exemple, un caractère chinois est décomposé en régions extrémités (E), lignes (R) et jonctions (S)⁶¹.

⁶¹[Cao 00] parle de régions régulières (R) et singulières (S).

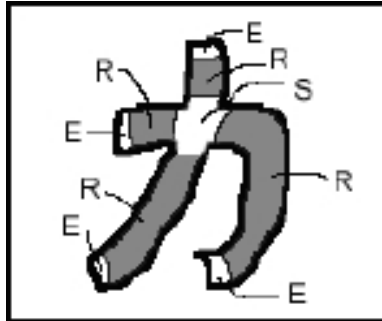


FIG. 1.31 – Segmentation en régions extrémités (E), lignes (R), et jonctions (S)

Il existe peu de travaux sur les méthodes de segmentation en régions. Nous n'avons référencé que ceux de [Chen 89] [Chen 93] [Chen 94] et [Cao 00]⁶². La problématique initiale sur ces méthodes a été introduite par [Pavlidis 86] et [Chen 89]. Ceux-ci exposent les divergences de résultats lors de l'utilisation de squelettiseur. La figure (1.32) illustre ces divergences à travers un exemple de [Chen 89]. Deux squelettiseurs parallèles différents⁶³ ont été employés sur un même caractère K . Ceux-ci fournissent deux résultats différents de squelettisation en terme de structure du squelette. Ces différences sont dues aux distorsions des jonctions introduites par ces deux squelettiseurs. [Pavlidis 86] en conclut : "C'est principalement un exemple de l'échec des techniques mathématiques à capturer le concept intuitif de forme. Ce sera le but des travaux futurs de réduire cette lacune en utilisant des informations globales"⁶⁴.

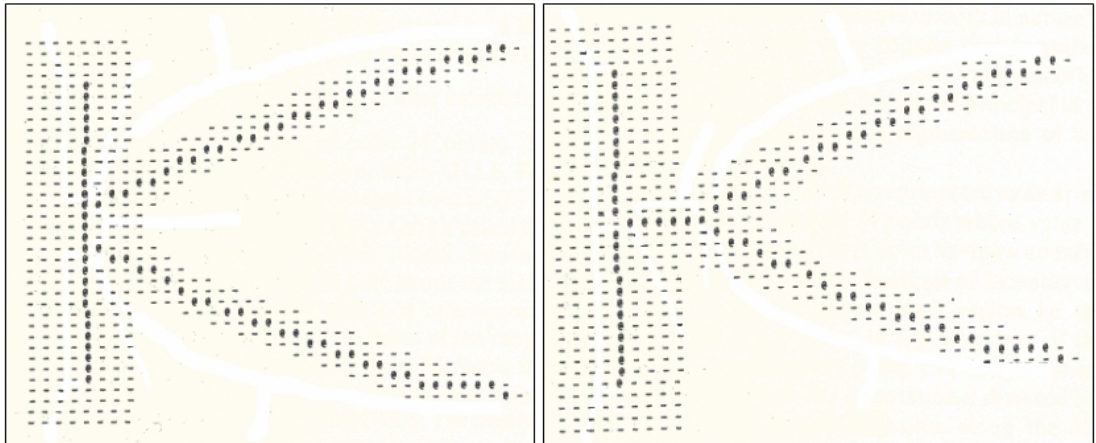


FIG. 1.32 – Deux résultats différents de squelettisation sur un même caractère

⁶²[Cao 00] propose des post-traitements améliorés à la méthode de [Chen 94].

⁶³Nous reportons le lecteur à [Chen 89] sur les squelettiseurs employés.

⁶⁴"It is mainly an example of the failure goal of mathematical techniques to capture the intuitive concept of shape. It will be the goal of future work to reduce this lability by using global information".

Ce principe a été repris et exploité par les méthodes de segmentation en régions. Celles-ci se servent des informations globales des régions lignes afin de vérifier la cohérence des régions jonctions extraites. Pour ce faire [Chen 94] calcule les distances d'orientation de l'image. Pour chaque pixel forme d'une image il trace des droites de projection passant par ce pixel de 0 à π . Ces droites étant tracées de part et d'autre des pixels les projections parcourent donc simultanément les voisinages $\{0, \pi\}$ et $\{\pi, 2 \times \pi\}$. L'écart d'orientation entre chacune des droites projetées est régulier et correspond au pas de précision de la méthode. Sur chacune des droites la distance d'orientation correspond à la distance Euclidienne séparant les deux pixels frontières les plus proches (figure (1.33)). Ces deux pixels frontières se trouvent donc aux intersections entre la droite projetée et les contours de la composante.

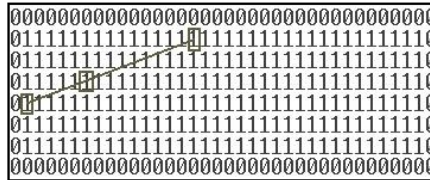
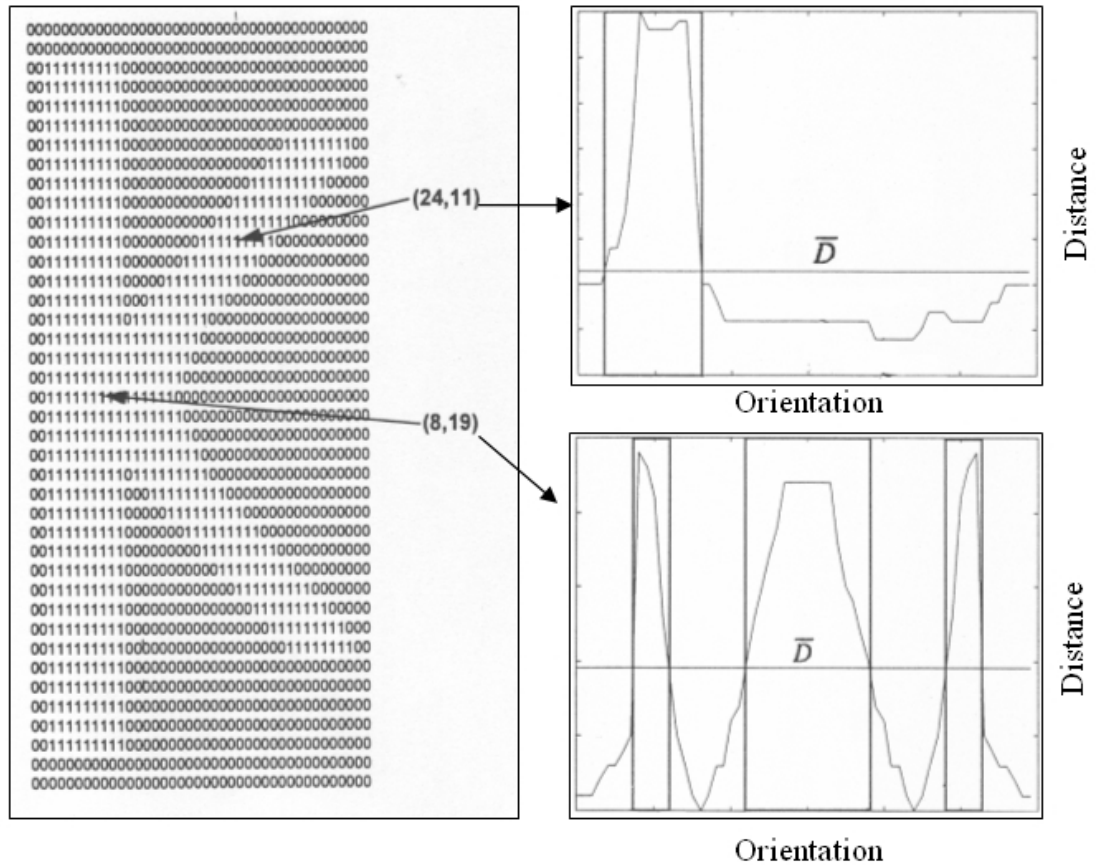


FIG. 1.33 – Calcul d'une distance d'orientation à $\frac{\pi}{6}$

Afin de distinguer les régions lignes et jonctions [Chen 94] analyse la courbe des distances d'orientation par rapport à la distance d'orientation moyenne. La figure (1.34) donne des exemples de courbes pour deux pixels du caractère K . La distance moyenne \bar{D} est obtenue par application de l'équation (1.20). Dans cette équation M représente le nombre de droite de projection tracée dans le voisinage de 0 à π . [Chen 94] recherche dans la courbe des distances d'orientation les pics de distance supérieurs à la distance d'orientation moyenne \bar{D} du pixel. Si plusieurs pics sont présents, le pixel appartient à une région jonction, sinon à une région ligne. Les pixels du caractère K de la figure (1.34) sont ainsi répartis en quatre régions lignes et une région jonction [Chen 94].

$$\bar{D}(i, j) = \frac{\sum_{m=0}^M D_m(i, j)}{M} \quad (1.20)$$

Les méthodes à base de segmentation en régions [Chen 94] [Cao 00] permettent ainsi de pallier aux problèmes de distorsion des jonctions rencontrés par exemple durant la squelettisation. Elles utilisent pour cela des informations globales aux jonctions [Pavlidis 86] (le voisinage des lignes) afin de guider leur processus de segmentation. Elles sont cependant de complexité algorithmique importante. En effet, pour chaque pixel forme plusieurs droites de projection sont calculées. Chaque droite implique le parcours de plusieurs pixels formes sur l'image. C'est pour cette raison que ces méthodes ne sont employées que sur des images de faibles tailles comme des imageries de caractère [Chen 94] [Cao 00].

FIG. 1.34 – Courbes des distances d'orientation de deux pixels du caractère *K*

1.7 Les méthodes à base de sous-échantillonnage

Les méthodes à base de sous-échantillonnage⁶⁵ ont été introduites par [Lin 85]. Elles ont été reprises par la suite dans les travaux de [Vaxivière 95]. Il n'existe depuis lors aucuns travaux complémentaires sur ces méthodes, elles sont donc peu utilisées [Wenyin 99]. Leur principe est de sous-échantillonner l'image en un ensemble de mailles⁶⁶. Ces mailles correspondent alors à des sous ensembles de pixels de l'image. Il existe plusieurs types de maille, [Song 00] présente les plus communes sur la figure (1.35). Cependant, le type carré (a) est celui communément utilisé [Lin 85] [Vaxivière 95]. Le but des méthodes à base de sous-échantillonnage est alors d'extraire des primitives graphiques à partir de l'analyse des mailles obtenues suite au sous-échantillonnage.

⁶⁵ subsampling

⁶⁶ meshes

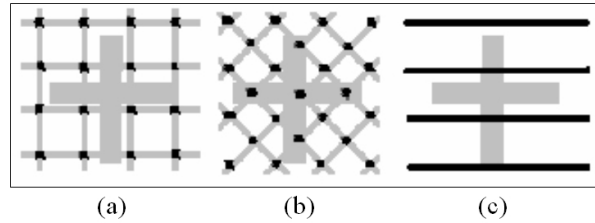


FIG. 1.35 – (a) mailles carrés (b) mailles losanges (c) mailles lignes

Une méthode à base de sous-échantillonnage se déroule en deux étapes principales [Vaxivière 95] : une de sous-échantillonnage et une de reconnaissance. L'étape de sous-échantillonnage parcourt l'image de façon à décrire les contours des mailles [Lin 85]. Ce parcours est donc fonction du type de maille adopté (figure (1.35)). À l'issue de ce parcours, les pixels décrivant les contours des mailles sont sauvegardés selon leur maille d'appartenance. Une fois les mailles extraites celles-ci sont reconnues par analyse de ces contours. La figure (1.36) illustre cette reconnaissance [Lin 85]. Celle-ci exploite les configurations des contours fonds et formes afin de reconnaître les mailles. Chaque configuration correspond alors à une maille définie dans une librairie connue a priori par le système.

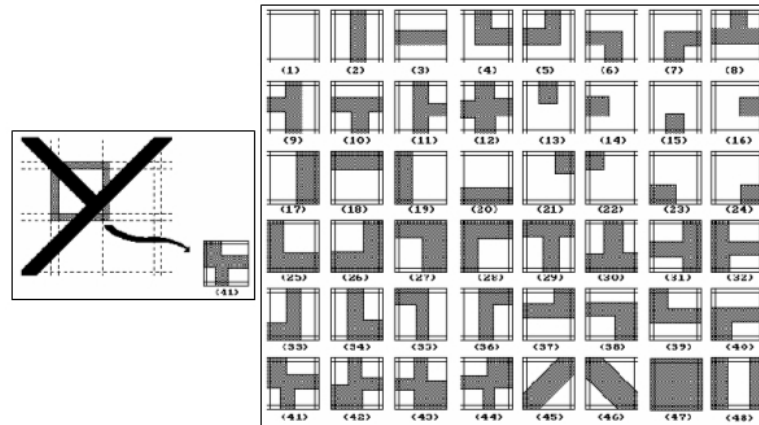


FIG. 1.36 – Reconnaissance des mailles de [Lin 85]

Les mailles reconnues sont ensuite exploitées afin d'extraire les primitives graphiques. Leur configuration est analysée afin d'extraire les relations topologiques entre mailles (connexion, parallélisme, ...) et d'en déduire les primitives existantes. La figure (1.37) illustre ce principe. À partir d'un ensemble de mailles reconnues (a) des vecteurs et jonctions ainsi que leurs relations de connexion sont extraits (b). Cet exemple illustre l'extraction de vecteurs verticaux/horizontaux mais d'autres primitives et attributs plus complexes peuvent être extraits. [Lin 85] [Vaxivière 95] extraient ainsi les vecteurs horizontaux/verticaux, pointillées, et obliques.

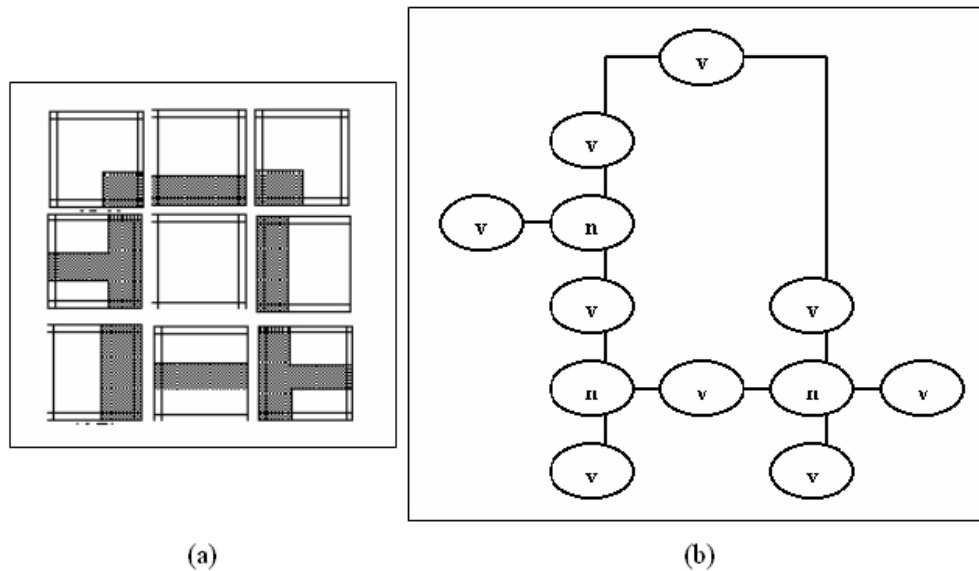


FIG. 1.37 – (a) mailles reconnues (b) primitives graphiques extraites⁶⁷

L'étape de reconnaissance des mailles s'appuie sur celle du sous-échantillonnage. Elle est donc dépendante des résultats de cette dernière. Un mauvais positionnement des mailles peut induire alors l'apparition de mailles non reconnues. Il peut alors induire des cas de sur-échantillonnage des mailles : apparition de plusieurs mailles pouvant être fusionnées. [Vaxivière 95] améliore la méthode de [Lin 85] afin de traiter ces problèmes de positionnement. Il utilise pour cela une procédure de correction en deux étapes de division/fusion⁶⁸ (figure (1.38)). La division (a) segmente une maille inconnue afin de la faire correspondre à un ensemble de mailles connues. La fusion (b) regroupe les mailles connues pouvant être représentées par une maille de plus haut niveau.

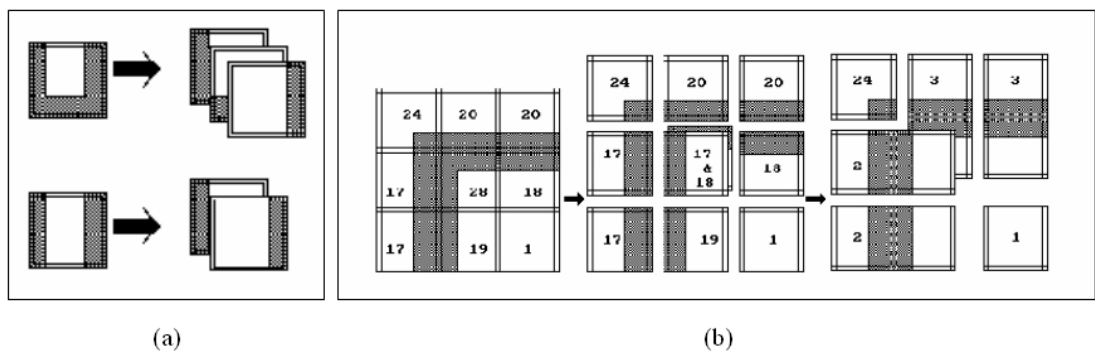


FIG. 1.38 – (a) division (b) fusion

⁶⁷ Les jonctions sont notées n et les vecteurs v .

⁶⁸ Nous les présentons sommairement ici et reportons le lecteur à [Vaxivière 95].

Les méthodes à base de sous-échantillonnage permettent une extraction rapide de primitives graphiques : seul un sous ensemble de l'image est analysé. Les primitives extraites sont de différentes natures : vecteurs, jonctions, Différentes relations topologiques peuvent également être extraites (connexion, parallélisme, orientation, . . .). À l'opposé des méthodes présentées précédemment dans ce chapitre, celles à base de sous-échantillonnage exploitent des modèles connus a priori. Elles utilisent une librairie de mailles préalablement définie décrivant les primitives graphiques des documents. Ces méthodes sont donc sensibles à l'adéquation existante entre les mailles décrites dans la librairie et les types de primitives graphiques présentes dans les documents. Elles sont de plus sensibles au positionnement initial des mailles. Elles nécessitent un ensemble de post-traitements complexes afin de pouvoir être utilisées efficacement.

1.8 Les méthodes à base de graphes de composantes

Les méthodes à base de graphes de composantes exploitent les composantes connexes et leurs relations topologiques comme primitives graphiques [Alnuweiri 92]. Les composantes connexes⁶⁹ sont des primitives graphiques fondamentales en analyse d'image [Rosenfeld 82]. [Shima 90] les définit de la façon suivante :

Définition 10 Composante connexe : *Les éléments d'une image, qui sont inter-connectés et ayant la même valeur, forme une composante connexe dans une image binaire*⁷⁰.

Les composantes connexes appartiennent à différentes catégories selon qu'elles composent le fond⁷¹ ou la forme⁷² d'une image de document [Ablameyko 00]. Les premières sont classiquement de couleur blanche et les secondes de couleur noire. La figure (1.39) illustre ces différentes catégories. Les composantes connexes composant la forme (a) sont celles majoritairement utilisées. Dans la littérature [Rosenfeld 82] [Shima 90] [Haig 92] [Suzuki 00] [Chang 04], les termes composantes et/ou composantes connexes sont d'ailleurs utilisés (par abus de langage) pour les désigner. Les composantes connexes composant le fond (b) se décomposent en deux catégories. On distingue premièrement la composante connexe directement connectée aux bords de l'image (c). Celle-ci est généralement nommée (également par abus de langage) de fond de l'image. Les autres composantes connexes composant le fond (d) sont qualifiées d'occlusions⁷³. Ces dernières sont caractérisées par des relations d'englobement (ou d'inclusion) vis à vis des composantes connexes composant la forme.

⁶⁹ connected components

⁷⁰ "Picture elements, that are connected to each other and have the same value, form a connected component in a binary image".

⁷¹ background

⁷² foreground

⁷³ holes or loops

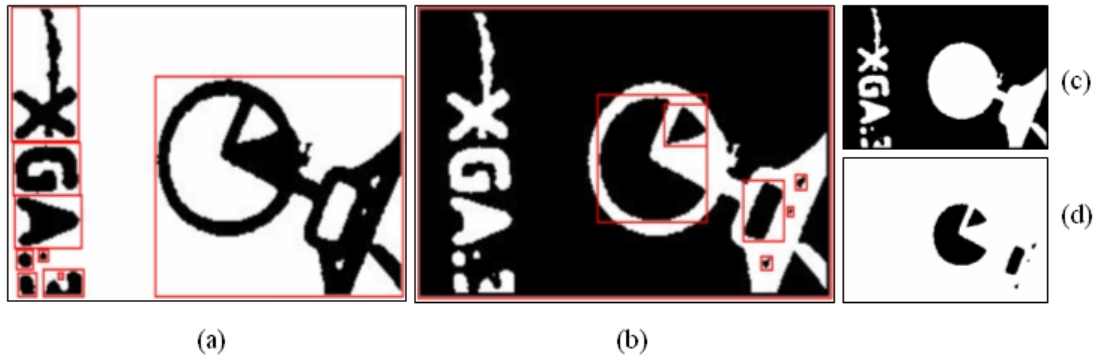


FIG. 1.39 – (a) composantes formes (b) composantes fonds (c) fond (d) occlusions

L'extraction des composantes connexes est basée sur l'utilisation de techniques de marquage de composantes connexes⁷⁴. Les recherches sur le marquage de composantes connexes ont débutées dans les années 1960 [Rosenfeld 66]. Elles se sont activement développées jusqu'au début des années 1990, de nombreux travaux et états de l'art y ont été consacrés [Kumar 91] [Alnuweiri 92]. Ces recherches se sont axées sur deux thématiques principales : algorithmes de marquage et architectures matérielles. En effet, différentes architectures matérielles (pipe-line, réseau maillé, massivement parallèle, ...) ⁷⁵ ont été développées pour la mise en oeuvre des algorithmes de marquage. Ces architectures optimisent la mise en oeuvre des algorithmes en temps de calcul, en capacités mémoire, ... Concernant les algorithmes de marquage [Shima 90] en référence trois catégories : par propagation, par balayage de lignes, et par suivi de contours. Ceux-ci marquent les différentes composantes connexes des images de document selon des critères de 4-connexité ou 8-connexité. Nous les présenterons par la suite.

Les algorithmes par propagation ont été parmi les premiers utilisés pour le marquage de composantes connexes [Rosenfeld 82] (figure (1.40)). Ils parcourent dans un premier temps l'image (a) à la recherche d'un pixel forme d'entrée (b). Ils marquent ensuite ce pixel et propagent (par analyse du 8 voisinage) son étiquette à l'ensemble de ses pixels voisins connexes (c). La propagation est ainsi répétée à partir des pixels marqués (d) tant que des pixels formes non marqués subsistent dans la composante connexe. Durant la propagation les caractéristiques topologiques des composantes ainsi que les pixels contours (non chaînés) internes ou externes peuvent être simultanément extraites. À l'issue du marquage de la composante connexe, l'algorithme reprend le parcours de l'image (b) à la recherche d'un pixel forme non marqué. Le parcours complet de l'image stoppe le marquage de l'image. Le marquage s'effectue donc en une passe sur l'image avec une analyse du 8-voisinage de chacun des pixels formes.

⁷⁴connected component labeling

⁷⁵Nous reportons le lecteur à [Kumar 91] et [Alnuweiri 92] sur ces aspects.

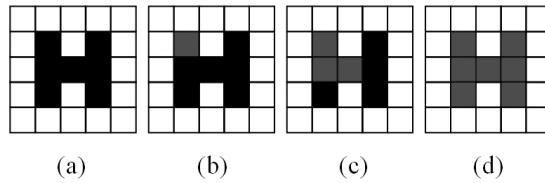


FIG. 1.40 – (a) image (b) pixel d'entrée (c) propagation 1 (d) propagation 2

Les algorithmes de marquage par balayage de lignes sont les plus répandus [Rosenfeld 82]. Ils ont été introduits par [Rosenfeld 66], depuis de nombreux travaux y sont consacrés [Agrawala 77], [Shima 90], [Helman 95], [Suzuki 00], [Wu 05], ... Ils emploient une approche comparable aux méthodes à base de décomposition en plages (section (1.5)). Ils encodent l'image sous forme de plages, les relations d'adjacence entre les plages sont alors utilisées pour déterminer leurs étiquettes. Le marquage s'effectue par un balayage en deux passes avant et arrière⁷⁶. Ces passes sont respectivement de sens lexicographique (\Rightarrow) et lexicographique inverse (\Leftarrow) [Ronse 04]. Durant la première passe l'image est encodée sous forme de plages horizontales. Les étiquettes des plages adjacentes sont alors propagées dans l'image (figure (1.41)) horizontalement (a) puis verticalement (b) à chaque balayage de ligne. Cette propagation des étiquettes applique sur chacun des pixels formes des masques 2 ou 4 voisinage selon que le marquage soit respectivement 4 ou 8 connexe. Durant ce balayage un tableau des étiquettes équivalentes est construit. La rencontre d'une relation d'adjacence entre plages d'étiquettes différentes (c) provoque une mise à jour de ce tableau (d). Ce tableau est ensuite utilisé durant la passe arrière (e) pour la mise à jour des étiquettes équivalentes. Les travaux récents sur ces algorithmes par balayage de lignes [Suzuki 00] [Wu 05] portent sur des méthodes d'optimisation pour leur mise en oeuvre⁷⁷.

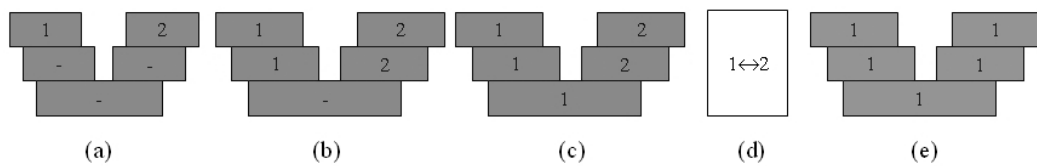


FIG. 1.41 – passe avant (a) (b) (c) tableau (d) passe arrière (e)

Les algorithmes de marquage par suivi de contours sont moins répandus que ceux par balayage de lignes [Shima 90]. Différents travaux y sont consacrés [Pavlidis 81], [Haig 92], [Zingaretti 98], [Chang 04], ... Ces algorithmes exploitent la détection de contours par suivi (section (1.2)) pour le marquage des composantes connexes. [Chang 04] en illustre le procédé sur la figure (1.42). Ils balayent les lignes de l'image de façon lexicographique (\Rightarrow) à la recherche des pixels formes. Lorsqu'un pixel forme

⁷⁶ forward and backward pass⁷⁷ Nous reportons le lecteur aux références citées sur ces aspects.

est rencontré (a) un suivi de contours est déclenché. Durant ce suivi les pixels contours de la composante connexe sont marqués. Une fois les contours marqués le balayage des lignes de l'image est repris. Durant le balayage d'une ligne, la rencontre d'un pixel contour marqué (b) provoque le marquage des pixels formes suivants jusqu'au prochain pixel contour marqué. Durant ce marquage, la rencontre d'un pixel contour englobé (c) déclenche un suivi de contours exploitant l'étiquette des contours englobants (d). Les contours englobés sont alors utilisés (e) durant le balayage des lignes (b) afin de compléter le marquage des pixels de la composante connexe.

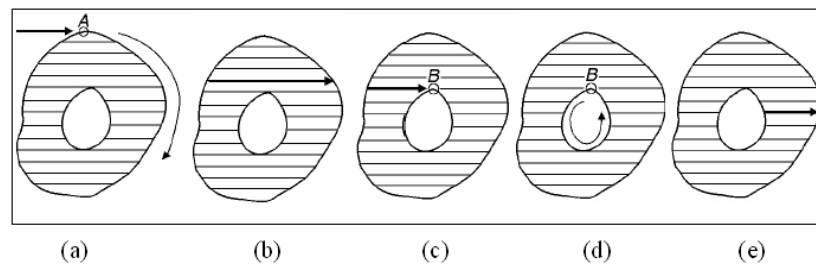


FIG. 1.42 – (a) suivi externe (b) marquage externe/externe
(c) (d) suivi interne (e) marquage interne/externe

Il existe peu de travaux comparatifs sur les algorithmes de marquage de composantes connexes [Chang 04]. Ceux par propagation sont de complexité importante dans le cas d'une forte densité de pixels formes sur l'image car ils en analysent le 8-voisinage. Ils présentent cependant l'avantage d'extraire simultanément au marquage les caractéristiques topologiques et les contours (non chaînés) des composantes. Ceux par balayage de lignes n'utilisent qu'une analyse du 4-voisinage des pixels formes mais exécute en contrepartie deux passes sur l'image. L'extraction des caractéristiques topologiques et des contours des composantes nécessitent cependant un sur-coût en mémoire et en temps de calcul. Les algorithmes par suivi de contours permettent de marquer les composantes connexes en une seule passe sur l'image plus le suivi des contours. Ils sont donc de plus faible complexité que les autres méthodes. Ils permettent de plus d'extraire et de chaîner les contours simultanément au marquage. Ces méthodes posent cependant des problèmes dans le cas des contours bruités (figure (1.43)). En effet, des occlusions 8-connexes du fond peuvent alors apparaître sur les bords (a). Celles-ci rendent le suivi ambigu : de multiples chemins peuvent être empruntés (b). Deux stratégies sont alors possibles. Soit la méthode adopte un suivi trigonométrique, dans ce cas les pixels contours les plus internes à la composante sont suivis. Les pixels contours les plus externes sont alors ignorés créant des problèmes de segmentation de ces pixels durant le marquage. À l'opposé la méthode peut adopter un suivi anti-trigonométrique, dans ce cas les pixels contours les plus externes à la composante sont suivis. Les pixels contours les plus internes sont alors ignorés alors que ceux-ci forment de meilleurs descripteurs de la composante. Les contours extraits sont alors d'avantage bruités.



FIG. 1.43 – (a) anti-trigonométrique (b) anti-trigonométrique vs trigonométrique

1.9 Comparaison des méthodes

Au cours des sections précédentes nous avons présenté les différentes méthodes existantes pour l'extraction de primitives graphiques à partir d'images de document. Elles se décomposent en différents niveaux d'extraction dans lesquels des primitives graphiques de bas niveau sont exploitées pour la construction de primitives graphiques de plus haut niveau [Song 02a]. Celles de plus haut niveau peuvent être encore par la suite exploitées par des méthodes⁷⁸ d'extraction à partir de primitives à base de vecteurs. Il existe de nombreuses similitudes entre ces différentes méthodes en ce qui concerne les primitives graphiques extraites, mais aussi les représentations employées pour décrire les formes graphiques. Le tableau (1.2) résume les principaux types de primitives graphiques : listes (de pixels), régions (composantes, plages, ...), et à base de vecteurs (vecteurs, arcs, courbes, ...). Le tableau (1.3) résume quant à lui les différentes représentations produites par les méthodes. Basé sur l'étude bibliographique de ce chapitre nous avons classé ces représentations en trois types : région (inclusion, voisinage, adjacence, parallélisme, ...), contour, et squelette (axes médians et jonctions). Celles-ci décrivent graduellement les formes graphiques de l'extérieur vers l'intérieur.

Primitives	Méthodes d'extraction
Listes	détection de contours, squelettisation, parcours des formes, décomposition en plages
Régions	détection de contours, squelettisation, parcours des formes, décomposition en plages, segmentation en régions, graphes de composantes
Vectorielles	parcours des formes, sous-échantillonnage

TAB. 1.2 – Types de primitives graphiques

Représentations	Méthodes d'extraction
Région	sous-échantillonnage, graphes de composantes
Contour	détection de contours, parcours des formes, décomposition en plages, segmentation en régions, graphe de composantes
Squelette	squelettisation, décomposition en plages, parcours des formes, sous-échantillonnage, segmentation en régions

TAB. 1.3 – Types de représentations

⁷⁸Nous n'avons pas abordé ces méthodes dans ce manuscrit (voir page 2).

Ces deux tableaux (1.2) et (1.3) illustrent les similitudes existantes entre les différentes méthodes d'extraction de primitives graphiques. Celles-ci permettent d'extraire des types communs de primitives. Elles permettent également de construire des représentations proches des formes graphiques des documents. Cependant elles emploient des approches différentes, chacune d'entre elles présente donc différents avantages et inconvénients. Certains travaux existent sur la comparaison de ces méthodes. Ceux-ci sont de type état de l'art [Tombre 00] [Wenyin 99] ou de type travaux d'évaluation [Phillips 99] [Song 03]. Ces travaux définissent cinq critères pour l'évaluation de ces méthodes : précision des jonctions, précisions des lignes, robustesse, complexité, réversibilité. Nous les détaillons dans la liste qui suit.

Précision des (J)onctions : Ce critère est développé par [Tombre 00], [Wenyin 99], et [Song 03]. Ceux-ci parlent respectivement de positionnement, de recouvrement, et de traitement des jonctions⁷⁹. Ce critère renseigne la capacité d'une méthode à détecter et à positionner correctement les jonctions d'une forme graphique.

Précision des (L)ignes : Ce critère est développé par [Tombre 00] et [Wenyin 99]. [Tombre 00] parle de critère de localisation : une méthode doit localiser correctement les lignes avec des erreurs de déplacement minimums. [Wenyin 99] parle quand à lui de qualité géométrique des lignes. Ce critère renseigne donc la capacité d'une méthode à détecter et à positionner correctement les lignes (ou axes médians) d'une forme graphique.

(Ro) bustesse : Ce critère est développé par [Tombre 00] et [Phillips 99]. [Tombre 00] parle de critère de détection. La méthode doit détecter les formes graphiques réelles et non des formes graphiques induites par le bruit. [Phillips 99] propose un protocole d'évaluation de la robustesse des méthodes basé sur quatre paramètres : mauvaise détection, non détection, fragmentation, et mauvaise fusion de primitives graphiques. Ce critère renseigne la capacité d'une méthode à résister au bruit sur les images de documents.

(C)omplexité : Ce critère est développé par [Wenyin 99] et [Song 03]. Il s'agit du critère de complexité algorithmique bien connu en informatique [Sedgewick 96]. [Wenyin 99] compare les différentes méthodes selon les complexités linéaire, quadratique, et cubique. [Song 03] effectue des tests d'exécution des différentes méthodes basés sur le nombre de passes nécessaires sur l'image.

(Re)stitution : Ce critère est développé par [Wenyin 99]. Celui-ci parle de préservation de l'épaisseur des lignes⁸⁰. Plus largement ce critère renseigne la capacité d'une méthode à restituer les formes graphiques d'entrée à partir des primitives graphiques extraites.

⁷⁹ junction positioning, recovery, and handling

⁸⁰ line width preservation

Selon notre point de vue trois critères supplémentaires peuvent être considérés : l'invariance, la sémantique, et la morphologie. Ces critères sont introduits dans les travaux de [Trier 96] et [Ramel 00]. Nous les détaillons dans la liste qui suit.

- (I)nvariance** : [Trier 96] dans son état de l'art développe ce critère en parlant d'invariance des méthodes aux changements d'échelles et d'orientations. Ce critère renseigne la capacité d'une méthode à résister aux variations topologiques des formes graphiques (translation, changements d'échelles et d'orientation, ...).
- (M)orphologie** : [Ramel 00] développe ce critère en évoquant la difficulté des méthodes à base de squelettisation à analyser les formes pleines. Il l'illustre à travers la figure (1.44). Ce critère renseigne la capacité d'une méthode à analyser tout type de forme graphique quelle que soit sa morphologie (linéaire, pleine, ...).
- (S)émantique** : [Ramel 00] développe ce critère en parlant de primitives graphiques de bas niveau et de haut niveau. Il considère alors les primitives graphiques à base de pixels comme de bas niveau par rapport à celles à base de vecteurs. Ce critère renseigne donc le niveau sémantique des primitives graphiques extraites par une méthode (liste de pixels, vecteurs, arcs, courbes, symboles, ...).

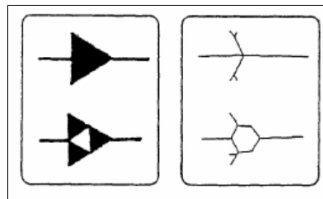


FIG. 1.44 – Problème de squelettisation avec les formes pleines

Basé sur ces huit critères nous comparons dans le tableau (1.4) les différentes méthodes d'extraction de primitives graphiques présentées dans ce chapitre selon leurs familles. Cette comparaison est basée sur les travaux existants de [Tombre 00] [Wenyin 99] et [Song 03] ainsi que sur le travail bibliographique présenté dans ce chapitre. Dans ce tableau nous avons adopté les notations de [Tombre 00] et [Wenyin 99] qui attribuent leurs critères selon trois niveaux (faible (-), bon (+), et excellent (++))⁸¹. Nous utilisons en complément la notation (\emptyset) lorsque la méthode ne permet aucune prise en compte d'un critère. Concernant le critère de complexité nous avons conservé à l'instar de [Song 03] une classification basée sur le nombre d'accès à la totalité des pixels de l'image⁸². Ce nombre d'accès dépend du nombre de passes sur l'image nécessaires aux méthodes, mais aussi des types de masques employés qui peuvent multiplier le nombre d'accès de 1 à 9⁸³. Nous détaillons par la suite ce tableau en synthétisant les avantages et inconvénients de chacune des méthodes d'extraction. Nous concluons alors sur la comparaison des méthodes.

⁸¹ Nous conservons ici la convention graphique de [Tombre 00].

⁸² Les traitements postérieurs des données extraites sont alors considérés comme négligeables.

⁸³ Les masques utilisés sont le plus souvent de type 3×3 .

Méthodes	J	L	Ro	C ⁸⁴	Re	I	M	S
Contours	∅	∅	++	1 à 2	++	++	++	–
Squelettisation	–	++	+	4	+	+	–	+
Parcours	+	+	+	1	∅	–	–	++
Plages	+	+	–	1 à 2	++	–	+	+
Régions	++	–	++	≫ 1	++	++	+	–
Mailles	+	+	–	≪ 0.5	∅	∅	–	++
Composantes	∅	∅	++	1 à 2	++	++	++	–

TAB. 1.4 – Comparaison des méthodes d'extraction de primitives graphiques

Les méthodes à base de détection de contours fournissent une représentation exacte des formes graphiques. Elles permettent donc l'analyse de tous les types de formes graphique, une restitution complète des images analysées à partir des contours extraits, et sont peu sensibles aux bruits. Elles sont de plus invariantes aux changements d'échelles et d'orientations. Elles sont enfin de faible complexité : celles à base de suivi de contours nécessitent une seule passe sur l'image et celles à base de morphologie deux passes (une pour la détection utilisant un masque (+) ou (×) sur les pixels formes, et une pour le chaînage). Leur inconvénient est la non détection des structures des formes graphiques (jonctions et lignes), ainsi que la faible sémantique des primitives graphiques extraites (listes de pixels).

Les méthodes à base de squelettisation permettent une bonne approximation des axes médians. Elles sont de plus invariantes aux changements d'orientations. Elles permettent également une restitution partielle⁸⁵ des images squelettisées. Leur inconvénient est le problème de distorsion des jonctions extraites. Elles sont de plus peu adaptées à l'analyse des formes pleines sur les images de document (seulement aux formes linéaires). Elles sont enfin de forte complexité, elles nécessitent trois passes sur l'image pour la squelettisation (deux pour la carte des distances et une pour l'axe médian) utilisant des masques 4-voisinage et 8-voisinage sur les pixels formes, plus une passe supplémentaire pour le chaînage.

Les méthodes à base de parcours des formes permettent une bonne détection des jonctions. De plus elles exportent des primitives graphiques à base de vecteurs et ne nécessitent qu'une seule passe sur l'image. Cependant, basées sur une recherche à priori de lignes elles sont peu adaptées à l'analyse de formes pleines. Elles ne permettent de plus aucune restitution des images analysées à partir des primitives graphiques extraites. Enfin les résultats d'extraction sont différents selon le sens de parcours adopté. Ces méthodes sont donc non invariantes aux changements d'orientations qui modifient le sens de parcours des formes graphiques.

⁸⁴Le (≪) et (≫) indiquent un nombre d'accès respectivement largement inférieur et supérieur.

⁸⁵La carte des distances calculée est de nature discrète.

Les méthodes à base de décomposition en plages permettent une bonne détection des jonctions et des lignes. Elles encodent de plus les formes graphiques sous forme de plages permettant une restitution complète des images de document analysées. Elles permettent également une bonne représentation des formes pleines sur les images de document. Elles sont enfin de faible complexité, elles ne nécessitent que de une à deux passes sur l'image (plages verticales et/ou horizontales) avec une analyse 4-voisinage sur les pixels forme pour la détection des relations de connexion entre plages. Elles sont cependant sensibles à l'adéquation existante entre les orientations des plages et celles des formes graphiques sur les images de documents. Elles sont donc peu adaptées pour l'analyse des formes inclinées. Elles sont de plus sensibles aux bruits impulsionsnels (ajout de petite composante fond) qui modifie la structure des plages encodées.

Les méthodes à base de segmentation en régions permettent une très bonne détection des jonctions. Elles utilisent pour cela les informations globales sur les lignes. Basées sur l'exploitation directe des pixels, elles sont robustes aux bruits, invariantes aux changement d'échelles et d'orientations, et permettent une restitution complète des images. Elles ne permettent par contre aucune extraction des axes médians (seulement la segmentation des régions lignes). Les primitives graphiques extraites sont de plus de faible niveau sémantique (régions jonctions et lignes). Enfin elles sont de complexité très importante, elles nécessitent pour chacun des pixels formes de l'image l'analyse de nombreux pixels voisins dans de multiples directions. Elles sont donc exploitables que sur des images de faibles dimensions.

Les méthodes à base de sous-échantillonnage permettent une extraction rapide de primitives graphiques. Seuls les pixels décrivant les contours des mailles sont parcourus sur l'image, et ceux-ci correspondent à un faible sous-ensemble de pixels de l'image. Les méthodes à base de sous-échantillonnage sont donc de complexité très inférieure aux autres méthodes. Basées sur l'analyse de ces contours elles permettent l'extraction de primitives graphiques complexes (vecteurs, jonctions, ...) mais aussi de différentes relations topologiques (connexion, parallélisme, orientation, ...). Ces méthodes ne permettent par contre aucune restitution de l'image à partir des mailles reconnues. Elles sont de plus dépendantes de l'adéquation existante entre les mailles décrites dans la librairie et les types de primitives graphiques présents dans les images de document. Ceci fait que ces méthodes ne sont pas invariantes aux changements d'échelles et d'orientations. Elles sont de plus sensibles au positionnement initial du maillage. Elles ne sont donc pas non plus invariantes aux translations. Elles sont également sensibles au bruit impulsionsnel : l'ajout de petites composantes formes sur l'image peut perturber la reconnaissance des mailles. Enfin, basées sur l'analyse de la configuration des contours des mailles elles sont peu adaptées à l'analyse des formes pleines.

Les méthodes à base de graphes de composantes sont très proches des méthodes à base de détection de contours. Elles ont donc des propriétés similaires. Elles diffèrent en ce qui concerne la complexité, elles sont à mi chemin entre les méthodes à base de détection de contours et celles à base de décomposition en plages. Les méthodes par balayage de lignes utilisent également en effet un encodage en plages. L'autre différence réside dans les primitives graphiques extraites qui sont de type région.

Au cours de cette section nous avons montré qu'il existait de nombreuses similitudes entre les différentes méthodes d'extraction de primitives graphiques. Elles permettent d'extraire des primitives graphiques ainsi que des représentations communes des formes graphiques présentes sur les images de document. Nous avons comparé ces différentes méthodes d'extraction selon différents critères. Aucune de ces méthodes ne semble parfaite : chacune d'entre elles possède divers avantages et inconvénients. Le choix entre les différentes méthodes existantes sera donc fonction de la nature des primitives graphiques à extraire, ainsi que de celles des images de document à analyser.

1.10 Conclusion

Au cours de ce chapitre nous avons présenté et comparé différentes méthodes d'extraction de primitives graphiques. Nous avons montré que celles-ci se décomposaient en différents niveaux d'extraction dans lesquels des primitives graphiques de bas niveau sont exploitées pour la construction de primitives graphiques de plus haut niveau. Nous avons également montré la complémentarité de ces méthodes. Chacune d'entre elles permet d'extraire un type de primitive graphique donné dans des conditions données (bruit, orientation, type de document, ...). Il y a donc un intérêt certain à combiner ces différentes méthodes pour l'extraction de primitives graphiques. Cette combinaison soulève cependant le problème de l'échange des primitives graphiques entre les différentes méthodes. Plus largement, cette combinaison soulève le problème de la gestion des connaissances dites graphiques au sein des systèmes d'analyse des documents. Nous présentons cette problématique dans le chapitre suivant.

Bibliographie

- [Ablameyko 93] S.V. Ablameyko & al. *System for Automatic Vectorisation and Interpretation of Graphics Images*. Pattern Recognition and Image Analysis, vol. 3, pages 39–52, 1993.
- [Ablameyko 00] S. Ablameyko & T.P. Pridmore. Machine interpretation of line drawing images. Springer Verlag Publisher, ISBN : 3-540-76207-8, 2000.
- [Adam 98] S. Adam. Interprétation de documents du réseau téléphonique, approche multi-spécialistes. Rapport de Master, Laboratoire PSI, Université de Rouen, France, 1998.
- [Agrawala 77] A.K. Agrawala & A.V. Kulkarni. *A Sequential Approach to the Extraction of Shape Features*. Computer Graphics and Image Processing (CGIP), vol. 6, no. 6, pages 538–557, 1977.
- [Ah-Soon 98] C. Ah-Soon. *Analyse des Plans Architecturaux*. Thèse de Doctorat, Institut National Polytechnique de Lorraine (INRIA), France, 1998.
- [Ahmed 02] M. Ahmed & R. Ward. *A Rotation Invariant Rule Based Thinning Algorithm for Character Recognition*. Pattern Analysis and Machine Intelligence (PAMI), vol. 24, no. 12, pages 1672–1678, 2002.
- [Alnuweiri 92] H.M. Alnuweiri & V.K. Prasanna. *Parallel Architectures and Algorithms for Image Component Labeling*. Pattern Analysis and Machine Intelligence (PAMI), vol. 14, no. 10, pages 1014–1034, 1992.
- [Arcelli 75] C. Arcelli, L. Cordella & S. Levialdi. *Parallel Thinning of Binary Pictures*. Electronic Letters, vol. 11, pages 148–149, 1975.
- [Badawy 02] O. El Badawy & M. Kamel. *Shape Representation Using Concavity Graphs*. In International Conference on Pattern Recognition (ICPR), pages 461–464, 2002.
- [Baja 94a] G.S. Di Baja. *Visual Form Representation*. In Human and Machine Vision : Analogies and Divergences, V, pages 115–129. Cantoni Editions, 1994.
- [Baja 94b] G.S. Di Baja. *Well-Shaped, Stable, and Reversible Skeletons from the (3,4)-Distance Transform*. Journal of Visual Communication and Image Representation, vol. 5, no. 1, pages 107–115, 1994.
- [Balaban 95] I.J. Balaban. *An Optimal Algorithm for Finding Segments Intersections*. In Symposium on Computational Geometry (SGC), pages 211–219, 1995.
- [Barrow 77] H.G. Barrow, J. Tenenbaum, R. Boles & H.Wolf. *Parametric Correspondence and Chamfer Matching : Two New Techniques for Image Matching*. In International Joint Conference on Artificial Intelligence (IJCI), pages 659–663, 1977.

- [Baruch 88] O. Baruch. *Line Thinning by Line Following*. Pattern Recognition Letters (PRL), vol. 8, no. 4, pages 271–276, 1988.
- [Bishnu 99] A. Bishnu & B.B. Chaudhuri. *Segmentation of Bangla Handwritten Text into Characters by Recursive Contour Following*. In International Conference on Document Analysis and Recognition (ICDAR), pages 341–344, 1999.
- [Black 81] W. Black & al. *A General Purpose Follower for Line Structured Data*. Pattern Recognition (PR), vol. 14, no. 1, pages 33–42, 1981.
- [Blostein 00] D. Blostein, E. Lank & R. Zanibbi. *Treatment of Diagrams in Document Image Analysis*. In Conference on Theory and Application of Diagrams (Diagrams), volume 1889 of *Lecture Notes in Computer Science (LNCS)*, pages 330–344, 2000.
- [Blum 67] H. Blum. *A Transformation for Extracting New Parameter of Shape*. Models for the Perception of Speech and Visual Form, pages 362–380, 1967.
- [Boatto 92] L. Boatto & al. *Detection and Separation of Symbols Connected to Graphics in Line Drawings*. In International Conference on Pattern Recognition, volume 2, pages 545–548, 1992.
- [Borgefors 01] G. Borgefors, G. Ramella & G.S. Di Baja. *Hierarchical Decomposition of MultiScale Skeletons*. Pattern Analysis and Machine Intelligence (PAMI), vol. 23, no. 11, pages 1296–1311, 2001.
- [Braun 95] A. Braun, T. Caesar, J.M. Gloger & E. Mandler. *Preprocessing Raw Binary Images by Means of Contours*. In International Conference on Document Analysis and Recognition (ICDAR), pages 640–643, 1995.
- [Bresenham 77] J. Bresenham. *A Linear Algorithm for Incremental Digital Display of Circular Arcs*. Communications of the ACM, vol. 20, no. 2, pages 100–106, 1977.
- [Burge 98] M. Burge & W.G. Kropatsh. *A Minimal Line Property Preserving Representation of Line Images*. In Conference on Structural and Syntactical Pattern Recognition (SSPR), volume 1451 of *Lecture Notes in Computer Science (LNCS)*, pages 355–368, 1998.
- [Cao 00] R. Cao & C.L. Tan. *A Model of Stroke Extraction from Chinese Character Images*. In International Conference on Pattern Recognition (ICPR), volume 4, pages 368–371, 2000.
- [Chai 92] I. Chai & D. Dori. *Orthogonal Zig Zag : an Efficient Method for Extracting Lines from Engineering Drawing*. In International Workshop on Visual Form (IWVF), pages 127–136, 1992.
- [Chang 04] F. Chang, C.J. Chen & C.J. Lu. *A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique*. Computer Vision and Image Understanding (CVIU), vol. 93, no. 2, pages 206–220, 2004.
- [Chen 89] Y.S. Chen & W.H. Hsu. *An Interpretive Model of Line Continuation in Human Visual Perception*. Pattern Recognition (PR), vol. 22, no. 5, pages 619–639, 1989.
- [Chen 93] Y.S. Chen. *An Improved Line Continuation Model in Human Visual Perception*. Pattern Recognition (PR), vol. 26, no. 6, pages 825–842, 1993.
- [Chen 94] Y.S. Chen. *Segmentation and Association Among Lines and Junctions for a Line Image*. Pattern Recognition (PR), vol. 27, no. 9, pages 1135–1157, 1994.

- [Chen 96] Y.S. Chen & Y.T. Yu. *Thinning Approaches for Noisy Digital patterns*. Pattern Recognition (PR), vol. 29, no. 11, pages 1847–1862, 1996.
- [Chiang 98] J. Chiang & S. Tue. *A New Algorithm for Line Image Vectorization*. Pattern Recognition (PR), vol. 31, no. 10, pages 1541–1549, 1998.
- [Coster 89] M. Coster & J.L. Chermant. *Précis d'analyse d'images*. Presses CNRS, ISBN : 2-87682-020-X, 1989.
- [Cowell 01] J. Cowell & F. Hussain. *Thinning Arabic Characters for Feature Extraction*. In Conference on Information Visualisation (IV), pages 181–185, 2001.
- [Deseilligny 98] M.P. Deseilligny, G. Stamon & C.Y. Suen. *Veinerization : A New Shape Description for Flexible Skeletonization*. Pattern Analysis and Machine Intelligence (PAMI), vol. 20, no. 5, pages 505–521, 1998.
- [Dori 99] D. Dori. *Sparse Pixel Vectorisation : An Algorithm and its Performance Evaluation*. Pattern Analysis and Machine Intelligence (PAMI), vol. 21, no. 3, pages 202–215, 1999.
- [Fan 98] K.C. Fan, D.F. Chen & M.G. Wen. *Skeletonization of Binary Images with Nonuniform Width via Block Decomposition and Contour Vector Matching*. Pattern Recognition (PR), vol. 31, no. 7, pages 823–838, 1998.
- [Ferreira 03] A. Ferreira, M.J. Fonseca & J.A. Jorge. *Polygon Detection from a Set of Lines*. In Encontro Português de Computação Gráfica (EPCG), pages 59–162, 2003.
- [Foskey 03] M. Foskey, M.C. Lin & D. Manocha. *Efficient Computation of A Simplified Medial Axis*. In Symposium on Solid Modeling and Applications (SMA), pages 96–107, 2003.
- [Freeman 61] H. Freeman. *On the Encoding of Arbitrary Geometric Configurations*. Transactions on Electronic and Computer, vol. 10, pages 260–268, 1961.
- [Haig 92] T.D. Haig, Y. Attikiouzel & M.D. Alder. *Border Following : New Definition Gives Improved Borders*. Vision Image and Signal Processing (VISIP), vol. 139, no. 2, pages 206–211, 1992.
- [Han 94] C.C. Han & K.C. Fahn. *Skeleton Generation of Engineering Drawings via Contour Matching*. Pattern Recognition (PR), vol. 27, no. 2, pages 261–275, 1994.
- [Hasan 00] M.Y. Hasan & L.J. Karan. *Morphological Reversible Contour Representation*. Pattern Analysis and Machine Intelligence (PAMI), vol. 22, no. 3, pages 227–239, 2000.
- [Helman 95] D. Helman & J. Jua. *Efficient Image Processing Algorithms on the Scan Line Array Processor*. Pattern Analysis and Machine Intelligence (PAMI), vol. 17, no. 1, pages 47–56, 1995.
- [Heutte 96] L. Heutte, J.V. Moreau, T. Paquet, Y. Lecourtier & C. Olivier. *Combining Structural and Statistical Features for the Recognition of Handwritten Characters*. In International Conference on Pattern Recognition (ICPR), pages 25–29, 1996.
- [Hilditch 69] C.J. Hilditch. *Linear Skeletons from Square Cupboards*. In Workshop on the Machine Intelligence, pages 403–420, 1969.
- [Jordan 97] C. Le Buhan Jordan & T. Ebrahimi. *Progressive Polygon Encoding of Shape Contours*. In Conference on Image Processing and its Applications (IPA), pages 17–21, 1997.

- [Kasturi 02] R. Kasturi, L. O’Gorman & V. Govindaraju. *Document Image Analysis : A Primer*. Sadhana, vol. 27, no. 1, pages 3–22, 2002.
- [Kégl 02] B. Kégl & A. Krzyzak. *Piecewise Linear Skeletonization Using Principal Curves*. Pattern Analysis and Machine Intelligence (PAMI), vol. 24, no. 1, pages 59–74, 2002.
- [Klette 03] G. Klette. *A Comparative Discussion of Distance Transforms and Simple Deformations in Image Processing*. Rapport technique CITR-TR-124, Computer Science Department, University of Auckland, New Zealand, 2003.
- [Kumar 91] V.K.P. Kumar. Parallel architectures and algorithms for image understanding. Academic Press, ISBN : 0125640404, 1991.
- [Labelle 98] L. Labelle, D. Lauzon, J. Konrad & E. Dubois. *Arithmetic Coding of a Lossless Contour-based Representation of Label Images*. In International Conference on Image Processing (ICIP), pages 261–265, 1998.
- [Lam 92] L. Lam, S.W. Lee & C.Y. Suen. *Thinning Methodologies - A Comprehensive Survey*. Pattern Analysis and Machine Intelligence (PAMI), vol. 14, no. 9, pages 869–887, 1992.
- [Lam 95] L. Lam & C.Y. Suen. *An Evaluation of Parallel Thinning Algorithms for Character Recognition*. Pattern Analysis and Machine Intelligence (PAMI), vol. 17, no. 9, pages 914–919, 1995.
- [Lanza 01] A. Lanza. *A Segmentation Algorithm for Line Drawings*. In Workshop on Graphics Recognition (GREC), 2001.
- [Lassaulzais 97] A. Lassaulzais. Contribution au développement d’un système de vectorisation et d’interprétation de plans de réseaux. Rapport de Master, Laboratoire PSI, Université de Rouen, France, 1997.
- [Lau 02] K.K. Lau, P.C. Yuen & Y.Y. Tang. *Stroke Extraction and Stroke Sequence Estimation On Signatures*. In International Conference on Pattern Recognition (ICPR), volume 3, pages 119–122, 2002.
- [Lefrere 93] L. Lefrere. *Contribution au Développement d’Outils pour l’Analyse Automatique de Documents Cartographiques*. Thèse de Doctorat, Université de Rouen, France, 1993.
- [Lin 85] X. Lin, S. Shimotsuji, M. Mihoh & T. Sakai. *Efficient Diagram Understanding with Characteristic Pattern Detection*. Computer Vision Graphics and Image Processing (CVGIP), vol. 30, pages 84–106, 1985.
- [Lin 95] J. Y. Lin & Z. Chen. *A Chinese Character Thinning Algorithm Based on Global Features and Contour Information*. Pattern Recognition (PR), vol. 28, no. 4, pages 493–512, 1995.
- [Lin 02] F. Lin & X. Tang. *Off-Line Handwritten Chinese Character Stroke Extraction*. In International Conference on Pattern Recognition (ICPR), volume 3, pages 249–252, 2002.
- [Liu 99] K. Liu, K., Y.S. Huang & C.Y. Suen. *Identification of Fork Points on the Skeletons of Handwritten Chinese Characters*. Pattern Analysis and Machine Intelligence (PAMI), vol. 21, no. 10, pages 1095–1100, 1999.
- [Lladós 01] J. Lladós, E. Martí & J.J. Villanueva. *Symbol Recognition by Error Sub-graph Matching Between Region Adjacency Graphs*. Pattern Analysis and Machine Intelligence (PAMI), vol. 23, no. 10, pages 1137–1143, 2001.

- [Lladós 02] J. Lladós, E. Valveny, G. Sánchez & E. Martí. *Symbol Recognition : Current Advances and Perspectives*. In Workshop on Graphics Recognition (GREC), volume 2390 of *Lecture Notes in Computer Science (LNCS)*, pages 104–127, 2002.
- [Loncaric 98] S. Loncaric. *A Survey of Shape Analysis Techniques*. Pattern Recognition (PR), vol. 31, no. 8, pages 983–1001, 1998.
- [Monagan 93] G. Monagan & R. Roosli. *Appropriate Base Representation Using a Run Graph*. In International Conference on Document Analysis and Recognition (ICDAR), pages 623–626, 1993.
- [Montoya 98] M.D.G. Montoya & I. Garcia. *Implementation of Parallel Thinning Algorithms on Multicomputers : Analysis of the Work Load Balance*. In Workshop on Parallel and Distributed Processing (PDP), pages 257–263, 1998.
- [Mullot 91] R. Mullot. *Segmentation d'Images et Extraction de Primitives pour la Reconnaissance Optique de Texte*. Thèse de Doctorat, Université de Rouen, France, 1991.
- [Murray 96] J.D. Murray & W. Van Ryper. *Encyclopedia of graphic file formats*. Editions O'Reilly, 2 edition, ISBN : 1565921615, 1996.
- [Nacken 93] P.F.M. Nacken. *Chamfer Metrics in Mathematical Morphology*. Rapport technique BS-R9309, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, Netherlands, 1993.
- [Nagy 92] G. Nagy. *At the Frontier of OCR*. IEEE, vol. 80, no. 7, pages 1093–1100, 1992.
- [Nieuwenhuizen 94] P.R. Van Nieuwenhuizen, O. Kiewiet & W.F. Bronsvort. *An Integrated Line Tracking and Vectorization Algorithm*. Computer Graphics Forum, vol. 13, no. 3, pages 349–359, 1994.
- [Nunes 00] P. Nunes, F. Marques, F. Pereira & A. Gasull. *A Contour-Based Approach to Binary Shape Coding Using a Multiple Grid Chain Code*. Signal Processing : Image Communication, vol. 15, no. 7-8, pages 585–599, 2000.
- [Ogier 94] J.M. Ogier. *Contribution à l'Analyse Automatique de Documents Cartographiques*. Thèse de Doctorat, Université de Rouen, France, 1994.
- [Paquet 92] T. Paquet. *Segmentation et Classification de Mots en Reconnaissance Optique de Textes Manuscrits*. Thèse de Doctorat, Université de Rouen, France, 1992.
- [Pavlidis 78] T. Pavlidis. *A Minimum Storage Boundary Tracing Algorithm and Its Application to Automatic Inspection*. Transactions on Systems, Man and Cybernetics (TSMC), vol. 8, no. 1, pages 66–69, 1978.
- [Pavlidis 81] T. Pavlidis. *Algorithms for graphics and image processing*. Computer Sciences Press, ISBN : 091489465X, 1981.
- [Pavlidis 86] T. Pavlidis. *A Vectorizer and Feature Extractor for Document Recognition*. Computer Vision Graphics and Image Processing (CVGIP), vol. 35, pages 111–127, 1986.
- [Phillips 99] I.T. Phillips & A.K. Chhabra. *Empirical Performance Evaluation of Graphics Recognition Systems*. Pattern Analysis and Machine Intelligence (PAMI), vol. 21, no. 9, pages 849–870, 1999.

- [Popel 02] D.V. Popel. *Compact Graph Model of Handwritten Images : Integration into Authentication and Recognition*. In Conference on Structural and Syntactical Pattern Recognition (SSPR), volume 2396 of *Lecture Notes in Computer Science (LNCS)*, pages 272–280, 2002.
- [Ramel 96] J.Y. Ramel. *Une Nouvelle Méthode pour l'Indexation Automatique de Dessins : Application aux Plans Cinématiques*. Thèse de Doctorat, Institut National Polytechnique de Lyon, France, 1996.
- [Ramel 00] J.Y. Ramel, N. Vincent & H. Emptoz. *A Structural Representation for Understanding Line-Drawing Images*. International Journal on Document Analysis and Recognition (IJ DAR), vol. 3, pages 58–66, 2000.
- [Ronse 04] C. Ronse. *Documentation de Traitement d'Images*. Université Louis Pasteur, France, 2004.
- [Rosenfeld 66] A. Rosenfeld & J.L. Pfaltz. *Sequential Operations in Digital Picture Processing*. Journal of the Association for Computing Machinery, vol. 13, no. 4, pages 471–494, 1966.
- [Rosenfeld 82] A. Rosenfeld & A.C. Kak. *Digital picture processing*. Academic Press, 2 edition, ISBN : 0125973020, 1982.
- [Rosin 97] P.L. Rosin. *Techniques for Assessing Polygonal Approximation of Curves*. Pattern Analysis and Machine Intelligence (PAMI), vol. 19, no. 6, pages 659–666, 1997.
- [Rutovitz 66] D. Rutovitz. *Pattern recognition*. Journal of the Royal Statistical Society, vol. 129, pages 504–530, 1966.
- [Sedgewick 96] S. Sedgewick & P. Flajolet. *Introduction à l'analyse des algorithmes*. International Thomson Publishing France, ISBN : 2841809579, 1996.
- [Shima 90] Y. Shima, T. Murakami, M. Koga, H. Yashiro & H. Fujisawa. *A High Speed Algorithm for Propagation-Type Labeling Based on Block Sorting of Runs in Binary Images*. In International Conference on Pattern Recognition (ICPR), pages 655–658, 1990.
- [Smith 04] M.J. Smith. *Distance Transforms as a New Tool in Spatial Analysis, Urban Planning and GIS*. Environment and Planning B : Planning and Design, vol. 31, no. 1, pages 85–104, 2004.
- [Sánchez 02] G. Sánchez, J. Lladós & K. Tombre. *An Error-Correction Graph Grammar to Recognize Texture Symbols*. In Workshop on Graphics Recognition (GREC), volume 2390 of *Lecture Notes in Computer Science (LNCS)*, pages 128–138, 2002.
- [Song 00] J. Song, F. Su, J. Cheng, C. Tai & S. Cai. *Line Net Global Vectorization : an Algorithm and its Performance Evaluation*. In Conference on Computer Vision and Pattern Recognition (CVPR), 2000.
- [Song 02a] J. Song, M. Cai, M.R. Lyu & S. Cai. *Graphics Recognition from Binary Images : One Step or Two Steps*. In International Conference on Pattern Recognition (ICPR), volume 3, pages 135–138, 2002.
- [Song 02b] J. Song, F. Su, C. Tai & S. Cai. *An Object-Oriented Progressive-Simplification based Vectorization System for Engineering Drawings : Model, Algorithm and Performance*. Pattern Analysis and Machine Intelligence (PAMI), vol. 24, no. 8, pages 1048–1060, 2002.

- [Song 03] J. Song, M.R. Lyu, M. Cai, & S. Cai. *Graphic Object Recognition from Binary Images : a Survey and Performance Comparison*. <http://appsrv.cse.cuhk.edu.hk/jqsong/>, 2003.
- [Suen 03] C.Y. Suen, S. Mori, S.H. Kim & C.H. Leung. *Analysis and Recognition of Asian Scripts - the State of the Art*. In International Conference on Document Analysis and Recognition (ICDAR), pages 866–878, 2003.
- [Suzuki 00] K. Suzuki, I. Horiba, & N. Sugie. *Fast Connected-Component Labeling Based on Sequential Local Operations in the Course of Forward Raster Scan Followed by Backward Raster Scan*. In International Conference on Pattern Recognition (ICPR), pages 2434–2437, 2000.
- [Taconet 90] B. Taconet, A. Zahour, S. Zhang & A. Faure. *Deux Algorithmes de Squelettisation*. In Colloque sur la Reconnaissance Automatique de l'Écriture (RAE), volume 68 of *BIGRE*, pages 68–78, 1990.
- [Thiel 01] E. Thiel. *Les Distances de Chanfrein en Analyse d'Images : Fondements et Applications*. Université de la Méditerranée, Aix-Marseille, France, 2001.
- [Tombre 96] K. Tombre. *Structural and Syntactic Methods in Line Drawing Analysis : To Which Extent Do They Work ?* In Conference on Structural and Syntactical Pattern Recognition (SSPR), volume 1121 of *Lecture Notes in Computer Science (LNCS)*, pages 310–321, 1996.
- [Tombre 00] K. Tombre, C. Ah Soon, P. Dosch, G. Masini & S. Tabbone. *Stable and Robust Vectorization : How to Make the Right Choices*. In Workshop on Graphics Recognition (GREC), volume 1941 of *Lecture Notes in Computer Science (LNCS)*, pages 3–18, 2000.
- [Trier 96] O.D. Trier, A. K. Jain & T. Taxt. *Features Extraction Methods for Character Recognition : A Survey*. *Pattern Recognition (PR)*, vol. 29, no. 4, pages 641–662, 1996.
- [Trupin 03] E. Trupin. *De la Reconnaissance Automatique d'Images de Documents*. Habilitation à Diriger les Recherches, Université de Rouen, France, 2003.
- [Turner 96] M.J. Turner & N.E. Wiseman. *Efficient Lossless Image Contour Coding*. *Computer Graphics Forum*, vol. 15, no. 2, pages 107–118, 1996.
- [Vaxivière 95] P. Vaxivière & K. Tombre. *Subsampling : A Structural Approach to Technical Document Vectorization*. In Conference on Structural and Syntactical Pattern Recognition (SSPR), pages 323–332, 1995.
- [Vinciarelli 02] A. Vinciarelli. *A Survey on Off-Line Cursive Word Recognition*. *Pattern Recognition (PR)*, vol. 35, no. 7, pages 1443–1446, 2002.
- [Wenyin 99] L. Wenyin & D. Dori. *From Raster to Vectors : Extracting Visual Information from Line Drawings*. *Pattern Analysis and Applications (PAA)*, vol. 2, no. 2, pages 10–21, 1999.
- [Wu 05] K. Wu, E. Otoo & A. Shoshani. *Optimizing Connected Component Labeling Algorithms*. In Conference on Image Processing, volume 5747 of *SPIE Proceedings*, pages 1965–1976, 2005.
- [Xue 01] H. Xue & V. Govindaraju. *Building Skeletal Graphs for Structural Feature Extraction on Handwriting Images*. In International Conference on Document Analysis and Recognition (ICDAR), pages 96–100, 2001.
- [Yoon 01] S. Yoon, G. Kim, Y. Choi & Y. Lee. *New Paradigm for Segmentation and Recognition*. In Workshop on Graphics Recognition (GREC), pages 216–225, 2001.

- [Yu 01] D. Yu & H. Yan. *Separation of Touching HandWritten Multi Numeral Strings Based on Morphological Structural Features*. Pattern Recognition (PR), vol. 34, no. 3, pages 587–599, 2001.
- [Zenzo 89] S. Di Zenzo & A. Morelli. *A Useful Image Representation*. In International Conference on Image Analysis and Processing (ICIAP), pages 170–178, 1989.
- [Zenzo 96] S. Di Zenzo, L. Cinque & S. Leviadi. *Run Based Algorithms for Binary Image Analysis and Processing*. Pattern Analysis and Machine Intelligence (PAMI), vol. 18, no. 1, pages 83–89, 1996.
- [Zhang 93] Y.Y. Zhang & P.S.P. Wang. *Analytical Comparison Of Thinning Algorithms*. International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), vol. 7, no. 5, pages 1227–1246, 1993.
- [Zhang 98] C. Zhang, S. Murai & E. Baltasvias. *Road Network Detection by Mathematical Morphology*. In Asian Conference on Remote Sensing (ACRS), pages 185–200, 1998.
- [Zhong 99] D.X. Zhong & H. Yan. *Pattern Skeletonization Using Run-Length-Wise Processing for Intersection Distortion Problem*. Pattern Recognition Letters (PRL), vol. 20, pages 833–846, 1999.
- [Zhu 99] S.C. Zhu. *Stochastic Jump-diffusion Process for Computing Medial Axis in MRF*. Pattern Analysis and Machine Intelligence (PAMI), vol. 21, no. 11, pages 1158–1169, 1999.
- [Zingaretti 98] P. Zingaretti, M. Gasparroni & L. Vecchi. *Fast Chain Coding of Region Boundaries*. Pattern Analysis and Machine Intelligence (PAMI), vol. 20, no. 4, pages 407–415, 1998.

Table des figures

Graphe de concavité d'un symbole	1
1.1 Méthodes d'extraction de primitives graphiques	3
1.2 Exemple de détection de contours	3
1.3 Exemples de résultats d'opérations morphologiques	4
1.4 Détection morphologique des contours internes et externes	5
1.5 Principe de l'étape de chaînage	6
1.6 Détection morphologique de contours proches	6
1.7 Détection de contours par suivi et par balayage	7
1.8 Squelette d'un extrait de plan de réseau	8
1.9 Squelettisation par amincissements successifs en 3 itérations	9
1.10 Éléments structurants non valides	11
1.11 Exemple de résultat de transformée de distance	11
1.12 8-graphe pondéré par masque de chanfrein (3×3)	12
1.13 Axe médian et boules maximales représentatives	14
1.14 Axe médian discret	14
1.15 Bruits typiques issus de la squelettisation	15
1.16 Graphe de squelette	16
1.17 Jonctions multiples	16
1.18 Détection des jonctions par suppression/reconstruction	18
1.19 Exemple de résultat de la méthode de parcours [Dori 99]	19
1.20 Éléments de parcours pixellaire/surfacique	20
1.21 Suivi de trait vectoriel	21
1.22 Suivi de trait circulaire	22
1.23 Suivi de trait surfacique	22
1.24 Détection vectorielle de [Lefrere 93]	24
1.25 Détection vectorielle de [Dori 99]	24
1.26 Détection surfacique de [Mullot 91]	25
1.27 Détection surfacique de [Ogier 94]	26
1.28 Décomposition en plages verticales	27
1.29 Graphe de plages	28
1.30 Segmentation d'un symbole par la méthode de [Boatto 92]	30
1.31 Segmentation en régions extrémités (E), lignes (R), et jonctions (S)	31
1.32 Deux résultats différents de squelettisation sur un même caractère	31
1.33 Calcul d'une distance d'orientation à $\frac{\pi}{6}$	32
1.34 Courbes des distances d'orientation de deux pixels du caractère K	33
1.35 Types de maille	34
1.36 Reconnaissance des mailles de [Lin 85]	34

1.37	Extraction des primitives graphiques à partir des mailles	35
1.38	Procédure de correction des mailles par division/fusion	35
1.39	Taxonomie des composantes connexes	37
1.40	Marquage par propagation	38
1.41	Marquage par balayage de lignes	38
1.42	Marquage par suivi	39
1.43	Problème d'ambiguïté avec le marquage par suivi	40
1.44	Problème de squelettisation avec les formes pleines	42

Liste des tableaux

1.1	Taxinomie des travaux de parcours de formes	19
1.2	Types de primitives graphiques	40
1.3	Types de représentations	40
1.4	Comparaison des méthodes d'extraction de primitives graphiques	43

Table des matières

Introduction première partie	0
1 Extraction de primitives graphiques, état de l'art	2
1.1 Introduction	2
1.2 Les méthodes à base de détection de contours	3
1.3 Les méthodes à base de squelettisation	8
1.3.1 Introduction	8
1.3.2 La squelettisation	8
1.3.3 La construction du graphe de squelette	15
1.4 Les méthodes à base de parcours de formes	19
1.4.1 Introduction	19
1.4.2 Suivi de traits	20
1.4.3 Détection de jonctions	23
1.5 Les méthodes à base de décomposition en plages	26
1.6 Les méthodes à base de segmentation en régions	30
1.7 Les méthodes à base de sous-échantillonnage	33
1.8 Les méthodes à base de graphes de composantes	36
1.9 Comparaison des méthodes	40
1.10 Conclusion	45
Bibliographie	46
Table des figures	54
Liste des tableaux	56