

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ AMAR TELIDJI LAGHOUAT



Faculté des Sciences

THÈSE présentée et soutenue par:

Hamini Nardjes

Le 14/12/2021

Pour l'obtention du diplôme de DOCTORAT en Sciences

Spécialité: Informatique

THÈME

Étude et amélioration des algorithmes de généralisation cartographique. Résolution du conflit d'empâtement et de la détection et typification de patterns.

Jury

LAGRAA Nasreddine	Professeur	(Université de Laghouat)	Président
TAHARI Abdelkarim	Professeur	(Centre universitaire d'Aflou)	Examineur
BELLAOUAR Slimane	MCA	(Université de Ghardaia)	Examineur
GUELLOUMA Younes	MCA	(Université de Laghouat)	Examineur
YAGOUBI Mohamed Bachir	Professeur	(Université de Laghouat)	Rapporteur

T  
H  
È  
S  
E

N° d'ordre: ...../2021-D/INF

# Remerciements

*Mes remerciements, avant tout, à **DIEU** tout puissant pour la volonté, la santé et la patience qu'il m'a donné durant toutes ces longues années d'études afin que je puisse arriver à ce stade.*

*Mes remerciements particuliers à mon promoteur, le Professeur Mohamed Bachir Yagoubi. Merci pour votre confiance et votre patience. La confiance que m'a témoigné le professeur Mohamed Bachir Yagoubi, le soutien et les conseils qu'il m'a prodigué tout au long de ce parcours de recherche, ainsi que l'autonomie qu'il m'a laissé, m'ont permis de réaliser cette thèse dans des conditions intellectuelles favorables au questionnement et à l'approfondissement de la pensée. Travailler avec lui est une expérience passionnante Il a toujours été là. C'est un homme que j'estime pour son honnêteté et son humanisme. J'espère pouvoir encore relever de nombreux autres défis avec lui dans les prochaines années.*

*J'exprime ma profonde reconnaissance à Monsieur, LAGRAA Nasreddine Professeur au département informatique pour l'honneur qu'il m'a fait d'accepter de présider le jury de cette thèse.*

*Je tiens à exprimer ma profonde gratitude envers Monsieur, GUELLOUMA Younes Professeur au département informatique, pour avoir accepté de juger ce travail.*

*Que Messieurs, TAHARI Abdelkarim, professeur au Centre Universitaire d'Aflou et BELLAOUAR Slimane, professeur à l'université de Ghardaia trouvent ici l'expression de mes plus vifs remerciements pour avoir accepté de participer au jury de soutenance et pour l'intérêt qu'ils ont bien voulu porter à ce travail.*

*Ma reconnaissance et mes remerciements vont également à Mes collègues et à tous ceux et celles qui ont contribué de près ou de loin, par leurs conseils, leurs suggestions et par leurs encouragements, à la réalisation de ce travail*

# Dédicace

*Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance, c'est tous simplement que : Je dédie cette thèse de doctorat à :*

*A Ma tendre Mère: Tu représente pour moi la source de tendresse et l'exemple de dévouement qui n'a pas cessé de m'encourager. Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études.*

*A Mon très cher Père: Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de vos sacrifices que vous avez consentis pour mon éducation et ma formation le long de ces années.*

*A mes très chères frère et sœurs : vos sacrifices, votre soutien moral et matériel m'ont permis de réussir mes études. Que ce travail soit un témoignage de ma reconnaissance et de mon amour sincère et fidèle.*

*Un remerciement spécial, profond et reconnaissant de tous les efforts fournis par ma chère sœur Meriem pendant toute la période de préparation de ma thèse. Je n'oublierai jamais tous ce que tu as fais pour moi.*

*A mes médecins et mes infirmières vos sacrifices votre dévouement, votre amour pour votre profession et votre générosité m'on permit d'être là aujourd'hui.*

*A tous mes enseignants depuis mes premières années d'études.*

*A tous ceux qui me sens chers et que j'ai omis de citer.*

# Résumé

La généralisation cartographique est un processus cartographique qui permet la production de cartes à plus petite échelle à partir de cartes à plus grande échelle. Car lorsque l'on réduit l'espace où se trouvent les objets cartographiques, des problèmes peuvent apparaître tels que la superposition, l'imperceptibilité, l'empâtement routier et l'encombrement de la carte. Il est nécessaire de sélectionner les informations à représenter sur les cartes. Il n'existe pas encore un outil de généralisation automatique complet du fait du manque d'algorithmes et de mesures de généralisation. C'est sur cette piste que nous avons travaillé et nous avons donc proposé un algorithme de généralisation de séries de virages routiers, une mesure de détection de patterns de bâtiments répétés, un algorithme de typification de patterns de bâtiments et un algorithme de calcul du diagramme de Voronoi pour les polygones, les polygones et les fonctions qui sont en cours de réalisation. Les algorithmes ont été implémentés sous PyQGIS et ont été testés sur des échantillons de données d'OpenStreetMap. Les résultats obtenus se sont révélés bénéfiques pour le domaine de la généralisation cartographique automatique.

**Mots-clés :** Généralisation-cartographique, automatisation, algorithme, route, empâtement, pattern, détection, typification, bâtiment, Voronoi.

# ملخص

التعميم الخرائطي هو عملية رسم خرائط تسمح بإنتاج خرائط بسلم أصغر من خرائط ذات سلم أكبر. لأنه عندما نقوم بتقليل المساحة التي توجد بها مجسمات الخريطة، يمكن أن تنشأ مشاكل مثل التداخل وعدم الظهور وانطباق الطريق على نفسها وازدحام الخريطة. من الضروري تحديد المعلومات التي سيتم تمثيلها على الخرائط. لا توجد حتى الآن أداة تعميم تلقائي كاملة بسبب نقص الخوارزميات ومقاييس التعميم. لقد عملنا على هذا المسار، ولذلك اقترحنا خوارزمية لتعميم سلسلة من انعطافات الطريق، وقياساً لاكتشاف أنماط البناء المتكررة، وخوارزمية لتصنيف أنماط البناء وخوارزمية لحساب الرسم التخطيطي لفورونوي للخطوط المتعددة، المضلعات والدوال التي هي في طور الإعداد. تم تنفيذ الخوارزميات تحت بيكوجس وتم اختبارها تحت عينات من أوبن سترت ماب وقد ثبت أن النتائج التي تم الحصول عليها مفيدة في مجال التعميم الآلي لرسم الخرائط.

**كلمات البحث:** التعميم الخرائطي، الآلية، الخوارزمية، الطرق، اندماج رمز طريق، النمط، الكشف، التصنيف، البناء، فورونوي

# Abstract

Cartographic generalization is a cartographic process that allows the production of smaller scale maps from larger scale maps. Because when we reduce the space where the map objects are located, problems can appear such as overlapping, imperceptibility, road coalescence and congestion of the map. It is necessary to select the information to be represented on the maps. A complete automatic generalization tool does not yet exist due to the lack of algorithms and generalization measures. It is on this way that we worked and we therefore proposed an algorithm for generalizing series of road turns, a measure for the detection of repeated building patterns, an algorithm for typifying building patterns and an algorithm for calculating the diagram of Voronoi for polylines, polygons and functions which is in preparation. The algorithms were implemented under PyQGIS and were tested on OpenStreetMap data samples. The results obtained have been shown to be beneficial for the field of automatic cartographic generalization.

**Keywords:** Cartographic-Generalization-, automation, algorithm, road, coalescence, pattern, detection, typification, building, Voronoi.

# Table des matières

<b>Chapitre 1 : Introduction</b> .....	<b>01</b>
1.1. Introduction .....	2
1.2. Problématique .....	7
1.3. Objectifs .....	8
1.4. Plan de travail .....	8
1.5. Contenu des chapitres .....	9
<b>Chapitre 2 : Généralités sur le spatial</b> .....	<b>11</b>
2.1. Introduction .....	12
2.2. L'information spatiale .....	12
2.3. Les données spatiales .....	13
2.4. Les systèmes de coordonnées .....	15
2.5. L'échelle d'une carte .....	16
2.6. Le processus de cartographie .....	16
2.7. Conclusion .....	18
<b>Chapitre 3 : La généralisation et son automatisation</b> .....	<b>19</b>
3.1 Introduction .....	20
3.2 L'automatisation de la généralisation .....	20
3.2.1 La généralisation automatique .....	20
3.2.2 La multi-représentation .....	21
3.2.3 La méthode combinée .....	23
3.3 Détail de la généralisation automatique .....	25
3.3.1 Le processus de généralisation .....	25
3.3.2 Les mesures de généralisation .....	25
3.3.3 Les algorithmes géométriques .....	25
3.3.4 Les opérateurs de généralisation .....	25
3.3.5 Les algorithmes de généralisation .....	28
3.3.6 Paramétrage des algorithmes .....	31
3.4 Synthèse .....	31
3.5 Synthèse sur les algorithmes de traitement des séries de virages .....	31
3.5.1 Analyse .....	34
3.6 Synthèse sur les mesures de détection des patterns de bâtiments .....	34
3.6.1 Analyse .....	35
3.7 Synthèse sur les algorithmes de typification de bâtiments .....	35
3.7.1 Typification de groupes structurés en pattern linéaire .....	35
3.7.2 Typification de groupes structurés en pattern grille .....	37
3.7.3 Typification du groupe non structuré .....	38
3.7.4 Analyse .....	39
3.8 Conclusion .....	39
<b>Chapitre 4: Proposition d'un nouvel algorithme pour les séries de virages (MBRS)</b> .....	<b>40</b>
4.1 Introduction .....	41

4.2	Qu'est-ce que l'empâtement? .....	41
4.3	Qu'est-ce qu'un virage? .....	42
4.4	Les formes de virages .....	42
4.5	Comment découper une route? .....	43
4.6	Algorithme Faille-Min pour les virages isolés .....	45
4.7	MBRS (Minimum-Break for Road Series) .....	46
4.7.1	Extraction des virages .....	48
4.7.2	Extraction du squelette .....	48
4.7.3	Extension des squelettes .....	50
4.7.4	Bufferisation .....	51
4.7.5	Découpe des buffers et extraction des virages résultants .....	52
4.7.6	Assemblage des virages .....	53
4.7.7	Réintégration de la série dans la route .....	55
4.8	Résultats .....	55
4.9	Comparaison .....	61
4.10	Synthèse et Conclusion .....	64
<b>Chapitre 5 : Proposition d'une mesure de détection de patterns de bâtiments</b>		
<b>répétés</b> .....		<b>65</b>
5.1	Introduction .....	66
5.2	Définition d'un pattern .....	66
5.3	La méthode .....	67
5.4	Résultats .....	71
5.5	Amélioration de la méthode .....	73
5.5.1	Analyse .....	76
5.6	Détails sur le diagramme de Voronoi .....	79
5.7	Synthèse et Conclusion .....	86
<b>Chapitre 6: Proposition d'un algorithme de typification de patterns de patterns</b>		
<b>de bâtiments</b> .....		<b>87</b>
6.1.	Introduction .....	88
6.2.	Définition de la typification .....	88
6.3.	Problématique de la typification des patterns de bâtiments .....	88
6.4.	La méthode .....	89
6.4.1.	Prétraitement .....	90
6.4.2.	Détection de la dispersion .....	91
6.4.3.	Sélection et positionnement .....	94
6.4.4.	Simplification des bâtiments .....	99
6.5.	Résultats .....	100
6.6.	Conclusion .....	104
<b>7. Conclusion générale</b> .....		<b>105</b>
7.1.	Conclusion .....	106
7.2.	Perspectives .....	107
<b>8. Bibliographie</b> .....		<b>109</b>

# Liste des figures

Figure 1.1 Plan de travail .....	09
Figure 2.1 Objets géographiques .....	13
Figure 2.2 Les données spatiales .....	14
Figure 2.3 Les types de projection .....	15
Figure 2.4 Echelle de carte .....	16
Figure 2.5 Le processus de cartographie .....	18
Figure 3.1 Modèle UML pour représentations multiples de Cosma .....	22
Figure 3.2 Modèle UML pour représentations multiples de Bobzien .....	23
Figure 3.3 Diagramme de classe UML représentant le patron géométrique .....	24
Figure 3.4 Classification des opérateurs de généralisation cartographique .....	27
Figure 3.5 Simplification de contours du bâti .....	28
Figure 3.6 Amalgamation .....	29
Figure 3.7 L'algorithme Plâtre .....	30
Figure 3.8 Typification de grille parfaite .....	37
Figure 3.9 Typification de quasi-grille .....	38
Figure 3.10 Fusion (unification) des sommets de Delaunay .....	39
Figure 4.1 Les effets de l'empâtement .....	41
Figure 4.2 Formes de virages .....	43
Figure 4.3 Détection des points d'inflexion .....	44
Figure 4.4 Transformation d'un virage en une surface .....	45
Figure 4.5 Application d'une triangulation de Delaunay sur le virage .....	45
Figure 4.6 Extraction de l'arbre squelette du virage .....	46
Figure 4.7 Choix du chemin le plus long comme squelette pour le virage .....	46
Figure 4.8 Les entrées de l'algorithme MBRS .....	47
Figure 4.9 extraction des virages .....	48
Figure 4.10 Extraction des squelettes .....	50
Figure 4.11 Extension des squelettes .....	51
Figure 4.12 Bufferisation .....	52
Figure 4.13 Virages résultants .....	53
Figure 4.14 Méthode d'assemblage de deux virages successifs .....	54

Figure 4.15 La série de virage résultante .....	55
Figure 4.16 Echantillons de routes .....	56
Figure 4.17 La mesure d'empâtement .....	57
Figure 4.18 Résultat n ° 1 .....	58
Figure 4.19 Résultat n ° 2 .....	59
Figure 4.20 Résultat n ° 3 .....	60
Figure 4.21 La mesure de l'enveloppe convexe .....	61
Figure 4.22 Comparaison .....	62
Figure 4.23 Problématique d'Accordéon .....	64
Figure 5.1 Exemples de Patterns de bâtiments .....	66
Figure 5.2 Le processus d'extraction de motifs.....	68
Figure 5.3 La méthode de groupage .....	70
Figure 5.4 La fonction de similarité .....	71
Figure 5.5 Résultat .....	72
Figure 5.6 Validation .....	73
Figure 5.7 Calcul de la matrice d'un groupe de bâtiments .....	74
Figure 5.8 Matrices de modèles particuliers .....	75
Figure 5.9 Des groupes de bâtiments .....	76
Figure 5.10 Des patterns de bâtiments .....	77
Figure 5.11 Graphique du taux d'erreur .....	78
Figure 5.12 Résultat 1 de Fortune sur une fonction sinus et un polygone .....	80
Figure 5.13 Résultat 1 de Voropix sur une fonction sinus et un polygone .....	81
Figure 5.14 Résultat 2 de Fortune (a) et de Voropix (b) sur une fonction sinus et un polygone .....	81
Figure 5.15 Voropix sur des points .....	82
Figure 5.16 Voropix sur des lignes .....	82
Figure 5.17 Voropix sur des polygones .....	83
Figure 5.18 Voropix sur des polygones .....	83
Figure 5.19 Courbes formées entre un point et la ligne de balayage .....	84
Figure 5.20 Courbes formées entre une ligne et la ligne de balayage .....	84
Figure 5.21 Courbes formées entre une polygones et la ligne de balayage .....	85
Figure 5.22 Courbes formées entre un polygone et la ligne de balayage .....	85
Figure 5.23 Courbes formées entre une sinus et la ligne de balayage .....	86

Figure 5.24 Fortune à gauche et Voropix à droite avec les différents types d'objets ...	86
Figure 6.1 Typification de bâtiments .....	88
Figure 6.2 Exemples de patterns de bâtiments .....	89
Figure 6.3 Etapes de la méthode .....	90
Figure 6.4 Division d'une zone en blocs .....	91
Figure 6.5 Détection des types de distributions spatiales des patterns de patterns .....	92
Figure 6.6 Extraction d'alignements .....	93
Figure 6.7 Détection d'une distribution de grille .....	94
Figure 6.8 Distribution de la formule de l'alignement avec différentes valeurs de m ...	95
Figure 6.9 Typification d'un pattern grille .....	96
Figure 6.10 Zone d'encombrement .....	97
Figure 6.11 Résultat de la fonction de coût .....	98
Figure 6.12 Résultat de l'algorithme de déplacement .....	99
Figure 6.13 Simplification du bâtiment .....	100
Figure 6.14 Application de la méthode sur un motif linéaire de motifs .....	101
Figure 6.15 Application de la méthode sur un motif de motifs en grille .....	101
Figure 6.16 Application de la méthode sur un motif de motifs non structuré .....	102
Figure 6.17 La mesure de l'étendue du terrain .....	103

# Liste des tableaux

Tableau 4.1: Comparaison de MBRS avec Accordéon et Schématisation .....	63
Tableau 5.1 Tests réalisés avec différents paramétrages de la taille de la matrice .....	76
Tableau 5.2 Nombre de cellules différentes selon la taille de la matrice de comparaison	77
Tableau 5.3 Taux d'erreur .....	78
Tableau 5.4 Comparaison entre les algorithmes du calcul du diagramme de Voronoi ...	79
Tableau 6.2 Mesure du résultat .....	103

# Abréviations

SIG : Système d'Information Géographique

MBRS : Minimum-Break for Road Series

GPS : Global Positioning System, en français : Système mondial de positionnement

GML : Geography Markup Language

UML : Unified Modeling Language

GEMURE : Généralisation multi-échelle et représentation multiple

GIS : Geographic Information System

QGIS : Quantum GIS

PYQGIS : Python QGIS

OLAP : Online Analytical Processing

SOLAP : Spatial OLAP

VUEL : View Element

# Chapitre 1 : Introduction générale

Ce chapitre est une introduction générale du sujet, nous y exposons notre problématique et nous y établissons nos objectifs. Un plan de travail détaille les étapes à suivre pour aspirer à atteindre les objectifs.

## 1.1. Introduction

L'émergence des nouvelles technologies a accru l'utilisation des cartes (routières, urbaines, touristiques, etc.). La demande de services de cartographie tels que la localisation, la planification d'itinéraire ou le trafic routier est en constante augmentation. Des outils comme Uber, Waze et Google Maps sont devenus incontournables pour le grand public. Le secteur professionnel n'est pas en reste avec des entreprises qui se développent à l'échelle mondiale comme les sociétés de livraison (Amazon, Aliexpress, Fedex, etc.), dont les patrons cherchent à suivre leurs employés (livreurs) en temps réel, et qui sont toujours à la recherche d'itinéraires plus rapides, et établissent des études pour découvrir les failles de leur systèmes pour gagner du temps et de l'argent. La cartographie concerne également des questions d'actualités telles que le suivi des populations, les épidémies et le changement climatique. La production de cartes est essentielle de nos jours. Mais produire une carte est un processus long et coûteux.

La première étape pour créer une carte est l'acquisition de données. Ces dernières années, avec les images satellites, les photographies aériennes et les images drones, on peut atteindre une précision de l'ordre de dizaines de centimètres. Après numérisation, nettoyage, mosaïquage et des traitements de photo interprétation, les cartes et les bases de données géographiques sont obtenues à très grande échelle (par exemple à l'échelle 1: 5000).

Mais les cartes qu'on achète dans les magasins ne sont pas à une grande échelle et peuvent tenir en poche. Il en va de même pour les échelles des applications utilisées à travers les technologies du numérique, par exemple, elles vont de 1: 5m à 1: 1000km dans le cas de Google maps. Cependant, les bâtiments de taille 20mx20m à une échelle de 1: 5m mesurent 4mmx4mm, et à une échelle de 1:25m, ils ne mesurent que 0,8mmx0,8mm. Une route d'une

largeur de 10 m ne fait que 0,05 mm à une échelle de 1: 200 000. La carte passe alors par la symbolisation puis la généralisation pour la rendre utilisable.

La généralisation est l'opération consistant à produire une carte à petite échelle à partir d'une carte à plus grande échelle [19, 56, 60]. Lorsque nous réduisons l'échelle de la carte, cela implique de réduire l'espace pour représenter les objets et de réduire les objets eux-mêmes. La carte est alors encombrée, l'information devient imperceptible et les parties de routes se fusionnent entre elles. La généralisation dispose d'une batterie d'outils pour mesurer et corriger tous ces problèmes comme la sélection des informations, l'amplification de taille et le déplacement pour n'en nommer que quelques-uns. La généralisation se fait par les cartographes de manière manuelle ou interactive. Lors de la généralisation, l'aspect de la carte et sa lisibilité sont privilégiés au lieu de la précision géographique. Pour corriger un chevauchement, la généralisation procède à un déplacement d'objets modifiant ainsi la position réelle des objets.

Les outils de la généralisation peuvent être regroupés en quatre grandes catégories: Les mesures cartographiques, elles fournissent des informations au processus de généralisation pour guider ses opérations, telles que la surface, la longueur, l'orientation, etc. Les algorithmes géométriques, comme le diagramme de Voronoi, la triangulation de Delaunay, l'enveloppe convexe, etc. Les algorithmes de généralisation, comme la simplification, le déplacement, la typification, la sélection, l'agrégation, etc. Enfin, le processus de généralisation qui est le chef d'orchestre qui gère l'ensemble des outils mentionnés précédemment pour effectuer la généralisation. Nous pouvons trouver des processus sous forme de modèle (liste établie d'étapes) et même des processus intelligents (réseaux de neurones, algorithme génétique, système à base d'agents, ...).

Il y a beaucoup de travaux sur l'automatisation de la généralisation mais comme le processus est laborieux et subjectif (car basé sur le sens artistique du cartographe), il n'y a pas de méthode entièrement automatisée. Cela est également dû à la non-disponibilité d'algorithmes avancés et de mesures de généralisation dans les outils SIG courants (ArcGIS, MapInfo, QGIS, FME).

Après étude et analyse des travaux existants, nous avons pu cibler le manque qui est dans le nombre d'algorithmes et de mesures de généralisation pour les différents types d'objets (routes, rivières, bâtiments, symboles ponctuelles) mais aussi au niveau d'un même type d'objets, car il est rare qu'il y ait un algorithme qui puisse traiter toutes les formes d'un même type d'objets. Le manque touche plusieurs types d'objets dont deux que nous avons décidé de traiter : les séries de virages et les patterns de bâtiments, ces deux éléments vont être décrits ci-dessous.

Dans notre travail, nous nous intéressons en premier lieu à la généralisation des routes. En effet, dès que l'on atteint l'échelle de 1: 200.000, les routes commencent à s'empâter (les parties de routes se fusionnent entre elles). L'empâtement est produit en raison de la symbolisation de la route qui attribue à chaque route un symbole, une couleur et une largeur supérieure à sa largeur réelle afin qu'elle puisse être reconnue. En raison de cette symbolisation, la route peut se chevaucher et sa forme devient incompréhensible.

Une route se compose de plusieurs parties, de zones linéaires, de virages isolés et de séries de virages. Les outils de généralisation des routes manquent, notamment en ce qui concerne les séries de virages. Il existe très peu de travaux sur l'étude des séries de virages en ce qui concerne leur généralisation [34, 51, 68, 88] et les travaux sur les types de séries de virages sont quasi inexistantes.

Dans ce document, nous présenterons une nouvelle approche pour traiter les séries de virages routiers pour résoudre l'empâtement [43]. Son principe est basé sur un algorithme testé et approuvé appelé Faille-Min [65]. A l'origine, cet algorithme a été créé pour gérer un virage isolé mais pas une série, le but de cette recherche est de l'adapter à cette dernière. Le choix de cet algorithme s'est basé sur sa capacité à corriger l'empâtement tout en minimisant l'emprise terrain du résultat, ceci est particulièrement utile lorsqu'il y a des objets dans les environs ou qu'il y a un espace limité pour modéliser la nouvelle série de virages. Le but par le choix de cet algorithme est également de pouvoir généraliser des formes de virages qui ne peuvent être traitées par les algorithmes existants.

Une simple application de cette méthode sur une série de virages ne fera pas l'affaire, car elle produira un résultat inutilisable. Pour une carte valide, un certain nombre de modifications sont nécessaires. Comme nous le verrons, Faille-Min produit des problèmes de jonction des virages résultants conduisant à la discontinuité de la ligne résultante, et l'harmonie des tailles des virages peut être altérée du fait de l'amplification des virages pour corriger l'empâtement. Nous allons donc corriger ces problèmes, c'est-à-dire que nous allons traiter les virages pour qu'ils s'assemblent d'une manière plus fluide, nous allons harmoniser leurs tailles, puis nous allons appliquer le nouvel algorithme que nous avons appelé MBRS (Minimum-Break for Road Series / Faille-Min pour les Séries de virages Routiers) sur des séries de virages issues de routes connues pour leur sinuosité (sujettes à l'empâtement) telles que les routes de montagne suivantes: Col de Chrea (Algérie), Mont Tianmen (Chine), Col Trollstigen (Norvège), Col de la Furka (Suisse) et Col de Turini (France). Nous procéderons ensuite à une comparaison avec les algorithmes existants de Schématisation [51] et d'Accordéon [68].

En second lieu nous nous intéressons à la détection et la typification de patterns de bâtiments. Les patterns représentent une propriété cartographique typique d'une zone qui doit être conservée après généralisation.

Dans la littérature, il existe de nombreux algorithmes qui visent à regrouper les bâtiments [82]. Nous disposons d'algorithmes de regroupement basés sur la proximité des bâtiments les uns par rapport aux autres [21]. Mais le risque avec ces algorithmes est de prendre dans le groupe de bâtiments appartenant à un modèle, des bâtiments qui n'y appartiennent pas, car il est difficile d'isoler les patterns du reste des bâtiments. Il existe également des méthodes qui identifient des patterns spécifiques tels que l'alignement ou la grille [69]. Enfin, nous trouvons des méthodes [97] qui identifient des modèles en comparant des groupes de bâtiments avec des modèles existant en mémoire et donc un modèle qui n'est pas dans la base de données ne peut pas être identifié. Il n'y a pas de méthode pour identifier des patterns quelconques qui se répètent comme le motif d'un tissu. Nous avons défini une nouvelle méthode pour résoudre ce problème basée sur la recherche de répétition de caractéristiques dans les groupes de bâtiments (critères similaires) [41]. Cette similitude consiste dans la forme, la taille, l'orientation et la position relative des bâtiments les uns par rapport aux autres, tandis que la méthode de regroupement est basée sur la recherche de proximité par un diagramme de Voronoï pour objets.

La typification de bâtiments permet de remplacer un ensemble de bâtiments par un autre ensemble avec un nombre moindre d'éléments mais qui le rend plus lisible, tout en conservant les caractéristiques du groupe d'origine (en ce qui concerne l'emprise terrain et la forme générale des bâtiments). Cependant, certains ensembles de bâtiments peuvent présenter des dispersions particulières (pattern) qu'il ne faut pas perturber comme un alignement le long d'une route, un agencement en grille, en cercle, en étoile, en lettre d'alphabet (Z, H, ...), ou n'importe quels bâtiments se faisant face de manière symétrique, il existe aussi des hiérarchies

de patterns (patterns disposés en pattern). Cette recherche concerne ce dernier type mais la méthode est applicable aux patterns simples aussi.

Notre méthode [42] passe par plusieurs étapes dont la détection de patterns répétés, la caractérisation de la dispersion, la sélection selon une fonction de coût. Les bâtiments sont ensuite simplifiés en appliquant une simplification de contour, une transformation ou une amplification en fonction de la taille du bâtiment. Enfin, nous induisons une opération de déplacement. La méthode a été implémentée sous le langage de programmation PyQGIS. Des exemples de cartes ont été prélevés dans l'état de Laghouat, en Algérie.

## 1.2. Problématique

Actuellement, il n'existe pas encore de solution de généralisation cartographique sur le marché. C'est-à-dire une application à laquelle on fournit une carte à grande échelle, à laquelle on fixe l'échelle désirée et qui nous fournit une carte généralisée à l'échelle voulue. La généralisation est encore réalisée pas à pas par les cartographes en utilisant des logiciels de cartographie mais de manière supervisée et humaine (processus humain). La raison est le manque d'algorithmes et de mesures de généralisation pour permettre au processus automatique d'avoir la vision et le talent d'un cartographe. Les problèmes généraux peuvent être regroupés dans les points suivants :

- Le manque d'algorithmes et de mesures de généralisation pour les différents types d'objets (routes, bâtiments, points thématiques, etc.).
- Le manque d'algorithmes et de mesures de généralisation pour les différentes variétés dans un même type d'objets.
- Le manque de travaux sur les processus de généralisation.

- La non-disponibilité des outils avancés de généralisation pour faciliter les études sur la généralisation.

Nous nous intéressons au manque d'algorithmes et de mesures de généralisation et nous avons pu mettre en évidence les deux sous-problèmes suivant :

- Manque de travaux pour l'étude et le traitement des séries de virages.
- Manque de travaux pour détecter et généraliser les patterns de bâtiments.

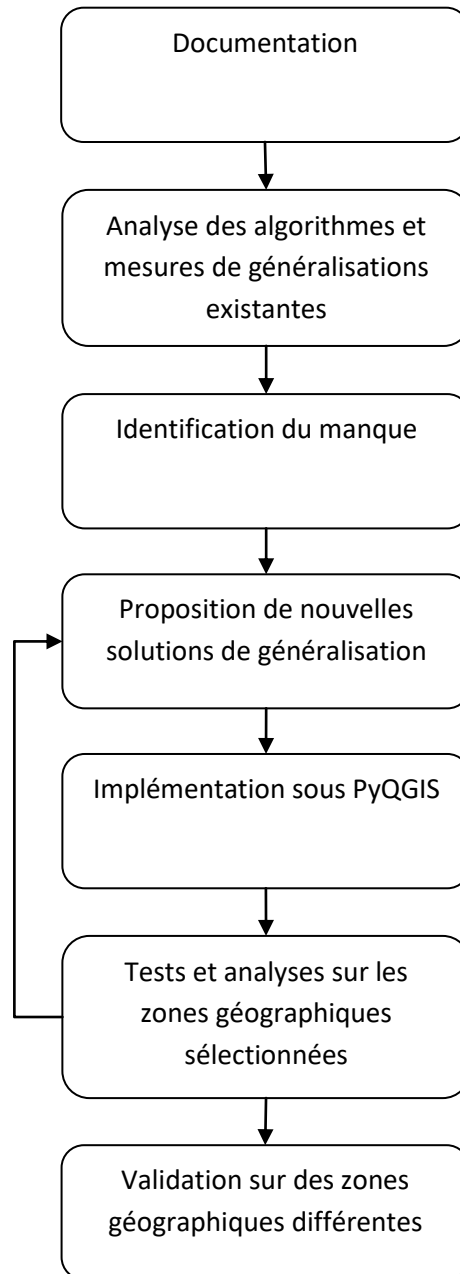
### **1.3. Objectifs :**

Notre objectif principal est de répondre au besoin de la généralisation automatique en algorithmes et mesures de généralisation. Et nos objectifs détaillés sont de proposer de nouveaux outils pour généraliser les séries de virages et les patterns de bâtiments, et sont résumés dans les points suivants :

- Proposition d'une méthode pour généraliser les séries de virages serrés basée sur l'algorithme Faille-Min.
- Proposition d'une méthode de détection des patterns de bâtiments répétés.
- Proposition d'une méthode de typification des patterns de bâtiments répétés.

### **1.4. Plan de travail**

Afin d'atteindre les objectifs énoncés précédemment, nous avons établis plusieurs étapes (Figure 1.1), dont une étude bibliographique et une analyse des algorithmes et mesures existants. Ensuite, nous avons émis une proposition de possibles méthodes pour répondre aux objectifs. Nous avons procédé à des tests et à des analyses des résultats afin de corriger les principes des méthodes proposées, jusqu'à validation du résultat.



**Figure 1.1 Plan de travail**

## **1.5. Contenu des chapitres**

Le présent document est divisé en sept parties. Le premier chapitre est une introduction générale du sujet, nous y exposons notre problématique et nous y établissons nos objectifs. Un plan de travail détail les étapes à suivre pour aspirer à atteindre les objectifs.

Le second chapitre est consacré au domaine du spatial, nous y définissons les aspects essentielles.

Le troisième chapitre présente un état de l'art sur la généralisation cartographique. Nous y exposons une étude comparative de l'existant en faisant mention du manque dans le domaine de la généralisation cartographique.

Au quatrième chapitre, nous présentons notre contribution dans le domaine du routier. Nous y exposons en détail notre algorithme de généralisation des séries de virages pour corriger l'empatement nommé MBRS (Minimum-Break for Road Séries).

Dans le cinquième chapitre, nous présentons notre contribution dans le domaine du bâti. Nous y détaillons donc une nouvelle mesure de détection des patterns de bâtiments. Nous y présentons également les résultats préliminaires de notre adaptation de l'algorithme de Fortune pour le calcul du diagramme de Voronoi pour les polygones, les polygones et les fonctions mathématiques. Le sixième chapitre est dédié à un nouvel algorithme de typification qui concerne lui aussi les patterns de bâtiments.

La dernière partie de ce document consiste en une conclusion générale suivie de nos perspectives dans le domaine de la généralisation cartographique automatique.

## Chapitre 2 : Généralités sur le spatial.

Dans ce chapitre, nous allons voir les aspects généraux de l'information spatiale, dont sa définition, sa représentation et sa localisation, en passant par les systèmes de coordonnées et le processus cartographique.

## 2.1. Introduction

Dans ce chapitre, nous allons voir les aspects généraux de l'information spatiale, dont sa définition, sa représentation et sa localisation, en passant par les systèmes de coordonnées et le processus cartographique.

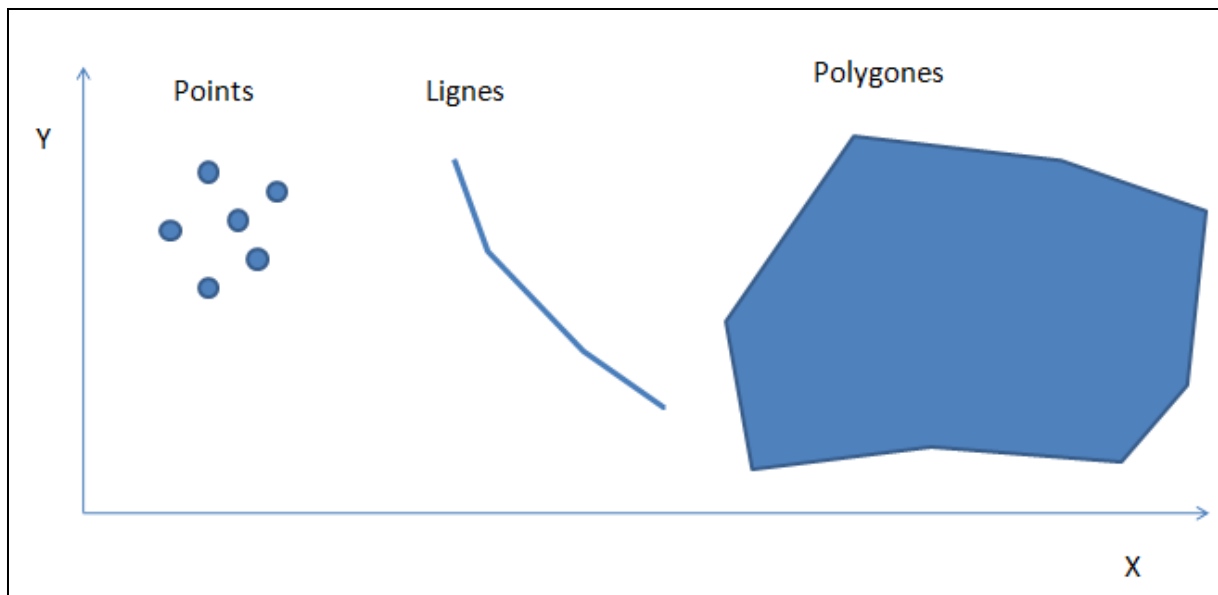
## 2.2. L'information spatiale

Le terme spatial désigne une étendue, un espace. Une information spatiale est une information qui porte une indication spatiale : une adresse, des coordonnées, un plan d'un bâtiment et bien entendu une carte.

Les objets sur la carte (Figure .2.1) sont représentés par : des polygones (pays, espaces verts, bâtiments), des lignes (routes, rails, rivières) et souvent dans des cartes à petite échelle par des points auxquels sont associés des pictogrammes (salles de sports, restaurants, hôtels, plages, hôpitaux, etc.).

Dans les bases de données, les polygones sont une suite fermée ordonnée de points reliés par des segments, une polyligne est une suite ouverte et ordonnée de points reliés par des segments, un point est une coordonnée X et une coordonnée Y.

Un objet de la carte est une association de trois informations : une géométrie (polygone, polyligne, point), une information graphique (couleur, épaisseur, motif, pictogramme pour le point, ...) et des attributs (nom, surface, nombre d'habitants, adresse).



**Figure 2.1 Objets géographiques**

### **2.3. Les données spatiales**

Les données spatiales peuvent être de sources différentes, nous en citons quelques-unes : Les images raster (tiff, png, bmp,..) issues d'images satellites, photos aériennes, drone. Les images raster ou matricielles se composent de pixels. Comme une matrice, chaque pixel est une cellule ayant son numéro de ligne et de colonne, et à laquelle est affectée une valeur de couleur selon la palette de couleur. Les images matricielles sont sujettes à la pixellisation lors d'un agrandissement ou un changement d'échelle.

Les images vecteur (shp, map) sont des images composées de géométries (polygone, polyligne, point). Elles ne sont pas sujettes à la pixellisation car l'image est recalculée à partir des coordonnées à chaque agrandissement.

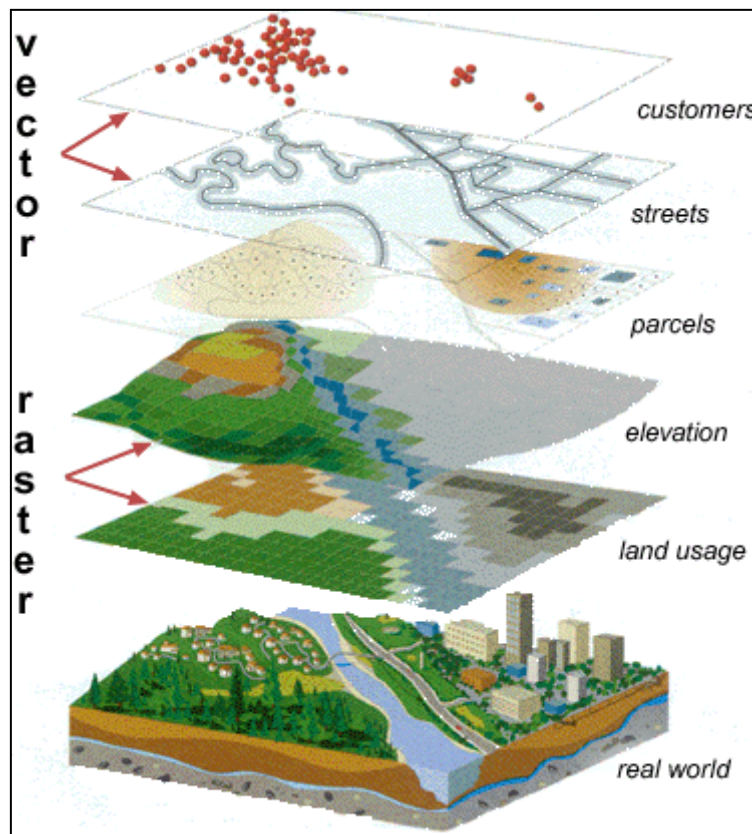
Grace aux constellations de satellites de géopositionnement, un appareil GPS peut relever des coordonnées ou des suites de coordonnées qui seront ensuite utiliser pour positionner des éléments sur la carte. Il est aussi possible de relever des coordonnées grâce à

la traditionnelle triangulation et aux réseaux géodésiques (des éléments de l'environnement dont la position est connue).

Les adresses, à travers le géocodage qui est une méthode qui permet de passer d'une adresse à la carte et de la carte à l'adresse.

Il y a bien d'autres sources de données comme la numérisation de plans et d'autres manières de localisation comme le gyroscope, ou des combinaisons des méthodes citées.

Les données dans une application spatiale (un système d'information géographique, un site web, une application mobile, etc.) sont organisées en couches (figure 2.2). Ces dernières sont superposées et ordonnées. Chaque couche représente une classe de données (routes, bâtiments, restaurants, hôtels, etc.).



**Fig2.2 Les données spatiales [78]**

## 2.4. Les systèmes de coordonnées

Il est possible de se situer sur le globe terrestre grâce aux systèmes de coordonnées (longitude latitude), il en existe plusieurs et ceci à cause de l'irrégularité du géoïde terrestre. Pour pouvoir effectuer les calculs, la terre est approximée à un ellipsoïde, chaque zone a son ellipsoïde qu'il lui convient. Pour passer du globe à la carte l'ellipsoïde est projeté. Il y a la projection cylindrique, le globe est entouré d'un cylindre et les points sur sa surface sont projetés (figure 2.3), le cylindre est ensuite déplié pour donner une surface plane. Il existe aussi la projection conique (cône) et azimutale (disque). Pour faire des relevés sur une carte, il est important de connaître son système de coordonnées.

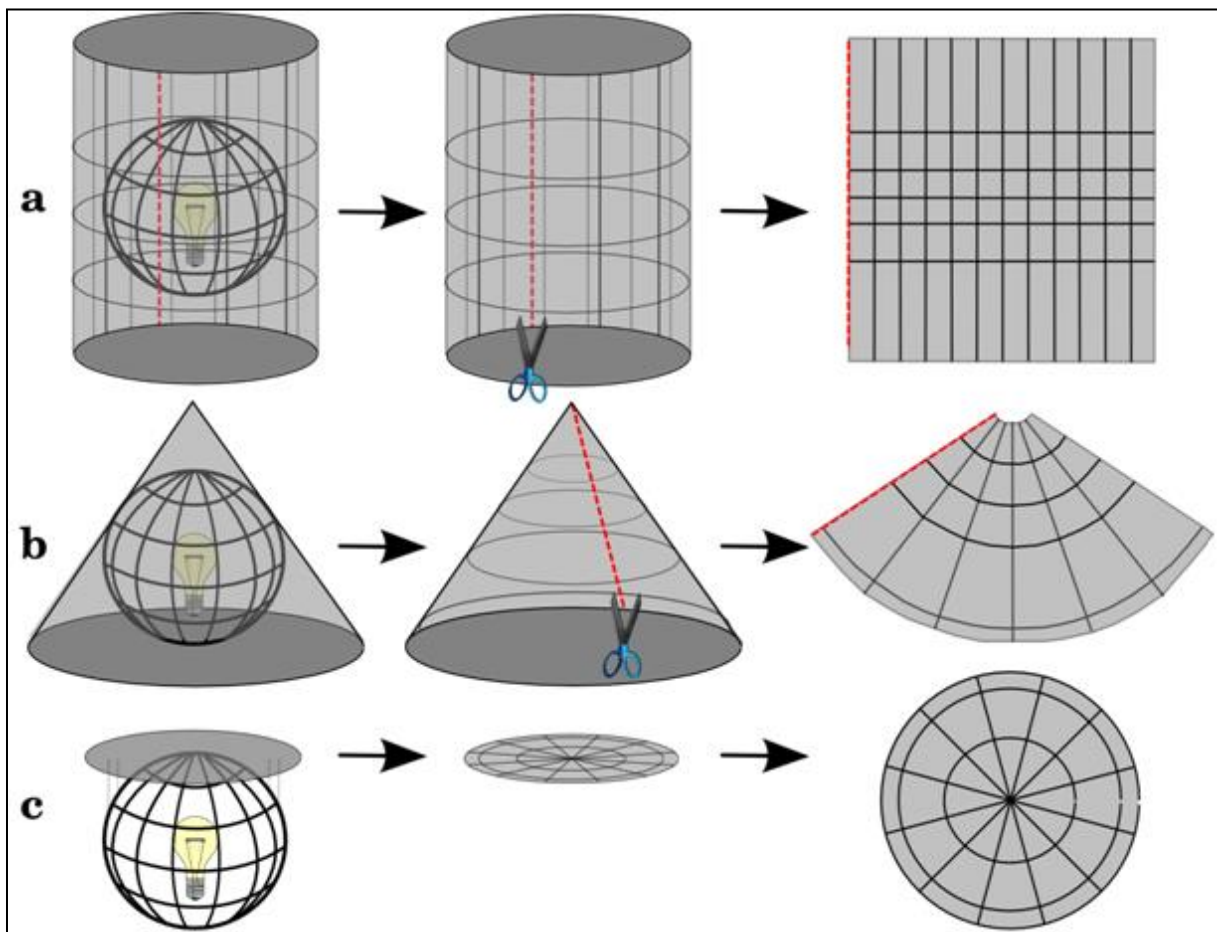


Figure 2.3 Les types de projection [71]

## 2.5. L'échelle d'une carte

L'échelle d'une carte (figure 2.4) représente le rapport entre la distance sur le terrain et la distance sur la carte. Par exemple 1: 20.000 veut dire qu'un centimètre sur la carte est égal à 200 mètres sur le terrain. De ce fait l'échelle 1:20.000 est plus grande que l'échelle 1:100.000.



Figure 2.4 Echelle de carte (GoogleMaps)

## 2.6. Le processus de cartographie

La réalisation d'une carte est un processus long et coûteux. Il requiert de combiner différentes sources de données afin d'offrir une carte précise qui permet à des personnes de se localiser ou de calculer des distances par exemple. En générale la production de carte passe par la photographie du territoire ciblé à travers des caméras de satellites (figure 2.5) ou des caméras à grande précision embarquées dans des appareils aériens.

Lors d'une aérophotographie les images dépassent les unes sur les autres afin de prendre des angles de vues différents, ce qui permet une meilleure appréciation des dimensions des objets sur le sol. Les images ensuite sont traitées, regroupées, localisées sur la

carte et mosaïquées. Vient ensuite l'étape de la vectorisation, c'est-à-dire passer de l'image matricielle contenant des pixels à une image vectorielle contenant des géométries.

Cela implique qu'on doit identifier les objets dans les images (photo-interprétation), et leur attribuer leurs véritables dimensions. C'est une étape délicate car une image prise à plat ne fournit pas la véritable longueur, hauteur ou largeur d'un objet. Pour ce faire les images qui se superposent sont mises côte à côte et grâce à la stéréoscopie on peut visualiser les images en relief (comme pour les films en 3D).

Ensuite on peut produire différentes cartes pour différentes thématiques ou différentes échelles. Si l'on veut produire par exemple une carte routière de poche. On doit sélectionner les informations pertinentes pour la thématique. Les géométries sont symbolisées (les routes colorées en rouge, les rivières en bleu, les espaces verts en vert). La symbolisation pour des effets visuels de lecture et de clarté peut par exemple attribuer à une route une épaisseur qui n'est pas la sienne, ce qui peut générer des superpositions avec les objets avoisinants ou un empâtement sur la même route. L'échelle doit être réduite afin de tenir en poche. L'espace pour représenter les objets est donc réduit, ils peuvent alors entrer en conflit les uns avec les autres et la carte peut paraître encombrée et illisible. Vient alors l'étape de la généralisation qui permet de corriger ces conflits.

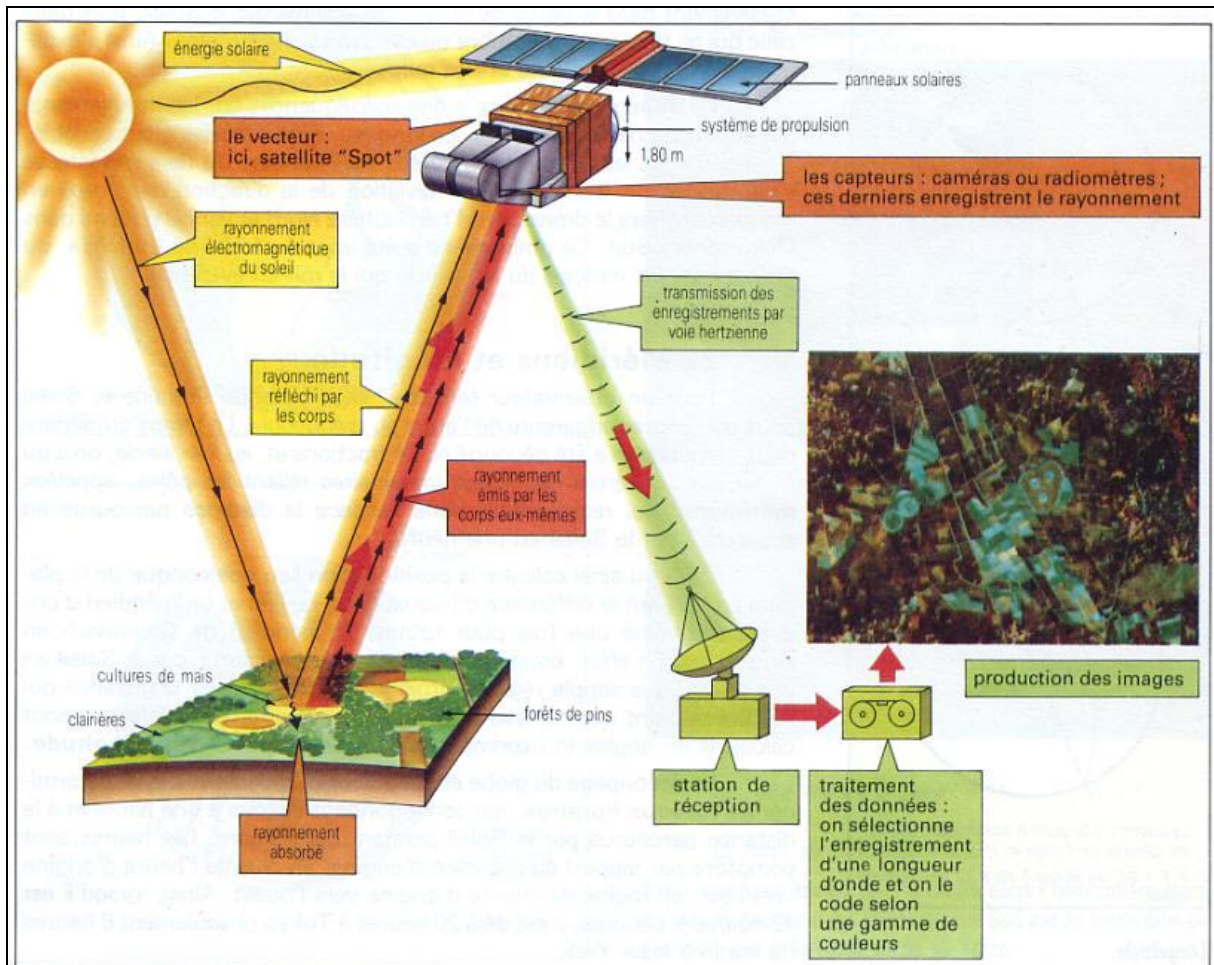


Figure 2.5 Le processus de cartographie [6]

## 2.7. Conclusion

Dans ce chapitre, nous avons vu quelques concepts du domaine du spatial. Le chapitre suivant est consacré à une des étapes de production de carte qui est la généralisation cartographique.

## Chapitre 3 : La généralisation et son automatisatisation

Ce chapitre présente un état de l'art sur la généralisation cartographique. Nous y exposons également une étude comparative de l'existant en faisant mention du manque dans le domaine de la généralisation cartographique.

### **3.1 Introduction**

La généralisation est une opération cartographique qui permet de produire des cartes à petites échelles à partir de cartes à grandes échelles [19]. Elle consiste à sélectionner les éléments importants à représenter à l'échelle déterminée. Elle procède aussi à la correction des conflits qui peuvent apparaître sur la carte : superposition, empâtement, imperceptibilité, encombrement, illisibilité. Et ceci en déplaçant les objets ou les parties d'objets superposés, en désempâtant les routes, en agrandissant les objets imperceptibles, en typifiant ou amalgamant des groupes d'objets du même type. Elle est habituellement réalisée par des cartographes expérimentés d'une manière supervisée. C'est un processus subjectif et artistique privilégiant la lisibilité à la précision géographique, de ce fait elle est difficile à automatiser.

### **3.2 L'automatisation de la généralisation**

Pour automatiser la généralisation, trois approches majeures sont dénombrées. La généralisation automatique, la multi-représentation et la méthode combinée.

#### **3.2.1 La généralisation automatique**

La généralisation automatique consiste à garder une carte à grande échelle en mémoire, les autres échelles sont générées automatiquement à partir de cette échelle [16, 22]. Différentes approches ont été tentées pour essayer de proposer une solution automatique qui pourrait remplacer la méthode manuelle. Obtenir des règles de généralisation à implémenter via interview des cartographes, mais cette méthode n'a pas donné des résultats précis, les experts n'étant pas habitués à décrire leur travail. On a essayé l'apprentissage supervisé [64] à partir d'échantillons de généralisation effectués manuellement, ça a permis de ressortir avec des règles de généralisations pour les routes et les bâtiments qu'on peut réutiliser. Cependant, les règles n'étaient pas assez suffisantes pour prendre en charge les multiples situations qui peuvent exister. Une autre approche a été tentée [66] visant à réaliser des plateformes de généralisation recueillant des algorithmes et des mesures de généralisation, ensuite proposer

des cartes à généraliser avec ces plateformes à des experts en généralisation. L'observation des choix d'outils des experts aurait permis d'avoir des règles à informatiser pour proposer une généralisation automatique. Cependant, les experts n'utilisaient pas les algorithmes implémentés pour le but qu'ils étaient faits pour, mais utilisaient des combinaisons d'outils pour réaliser la tâche à faire. Ils n'étaient pas non plus satisfaits du résultat des algorithmes. Des modèles de généralisation ont été proposés [64, 75] qui consistent en une liste d'étapes à suivre pour aboutir à la généralisation mais trop généraux. Des approches du domaine de l'intelligence artificielles ont été tentées comme les systèmes multi-agents [29, 35] permettant un traitement local par des agents micro (bâtiments, routes) et une gestion globale par des agents méso (cartier, zone, ...). Les algorithmes génétiques [25, 40, 90, 95] qui permettent de tester plusieurs solutions et de converger vers la solution la plus optimale mais les travaux sont peu nombreux et le problème de ces méthodes est le temps de réponse. Cependant, les progrès technologiques pourraient réduire le temps de réponse de ces méthodes.

### **3.2.2 La multi-représentation**

La multi-représentation consiste à garder en mémoire les différentes échelles de carte et à afficher à la demande la carte correspondante à l'échelle demandée. Il y a plusieurs recherches pour proposer des structures de base de données pour accueillir ces représentations, il y a Cosma [23] qui a proposé un modèle basé sur GML (Geography Markup Language) pour le web ( Figure 3.1).

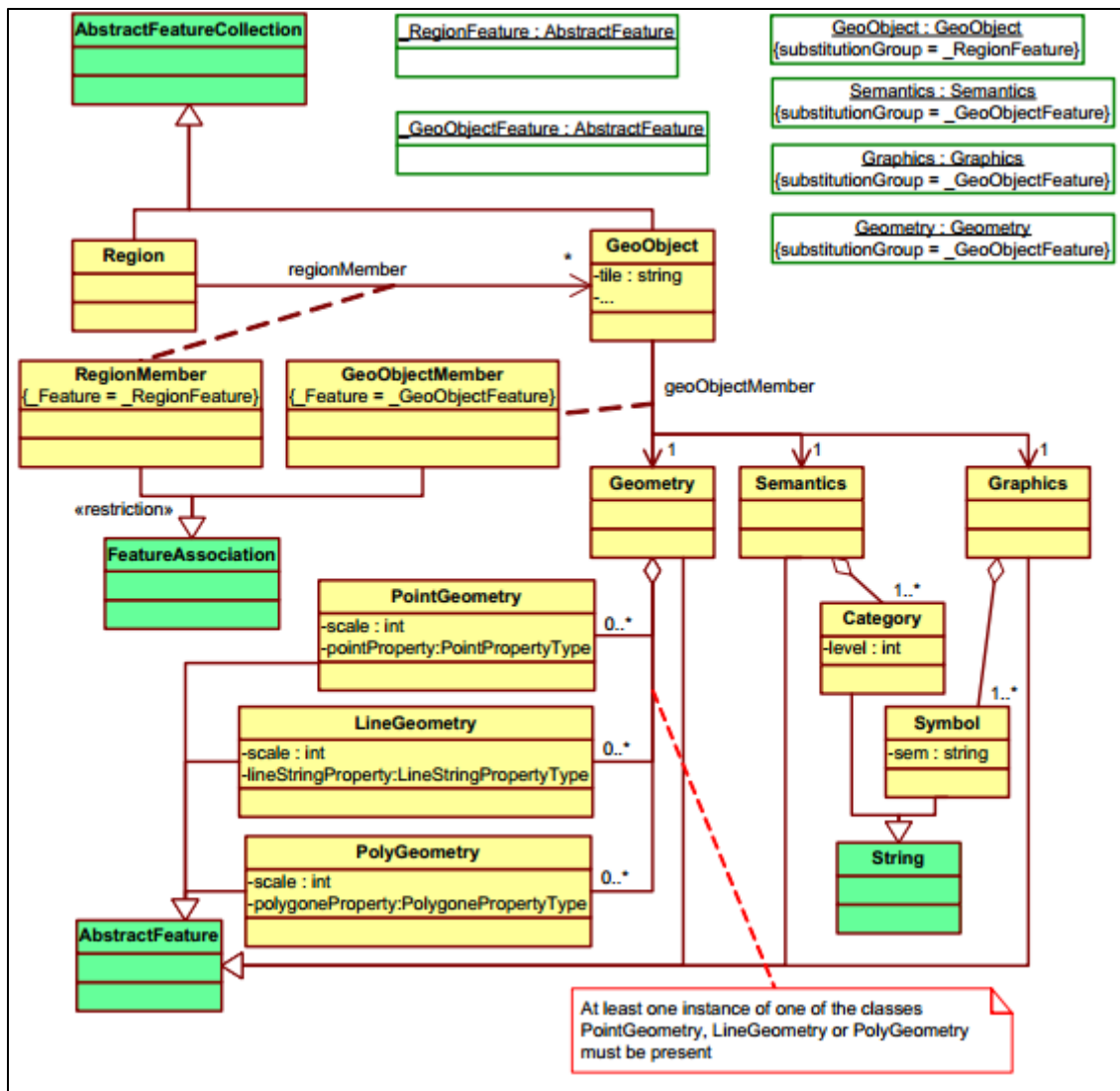


Figure 3.1 Modèle UML pour représentations multiples de Cosma [23]

Il y a des travaux qui proposent une structure pour base de données transactionnel [14, 15] (Figure 3.2). On a aussi des approches qui proposent une structure spécialisée pour entrepôt de données spatial nommé Vuel (View Element) [14]. Cependant l'espace de stockage engendré par les représentations est important, s'ajoute à cela le problème des mises à jour d'un objet. Quand celui-ci possède de nombreuses représentations dans la base de données, cette tâche s'avère complexe.

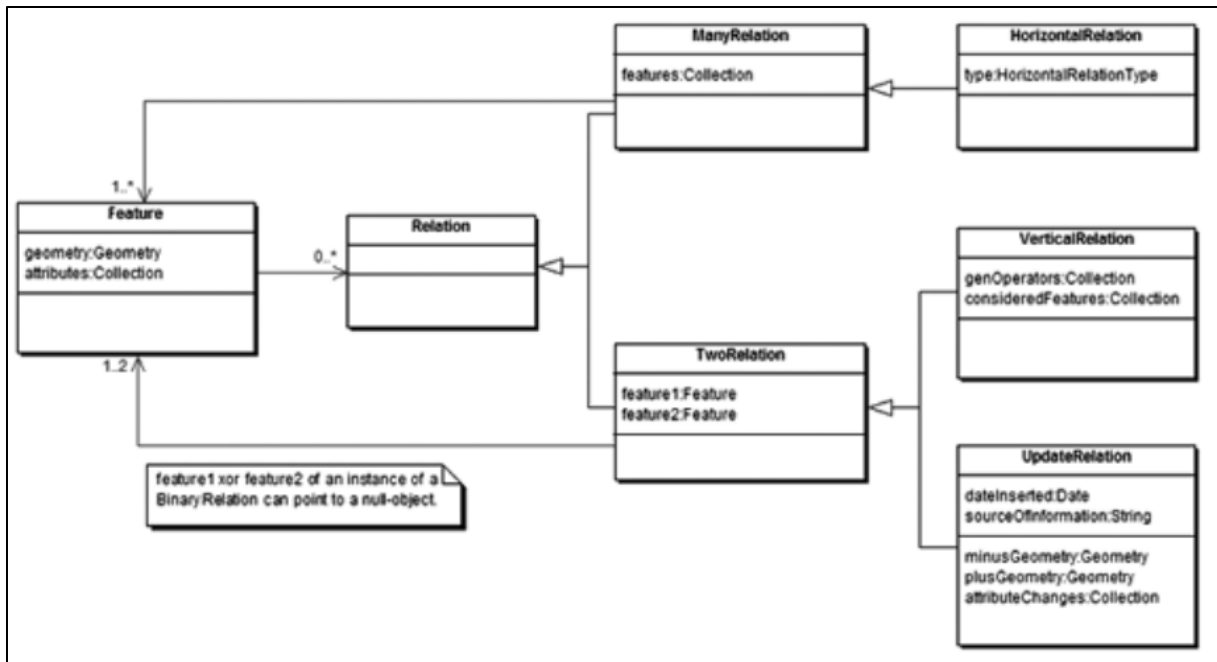
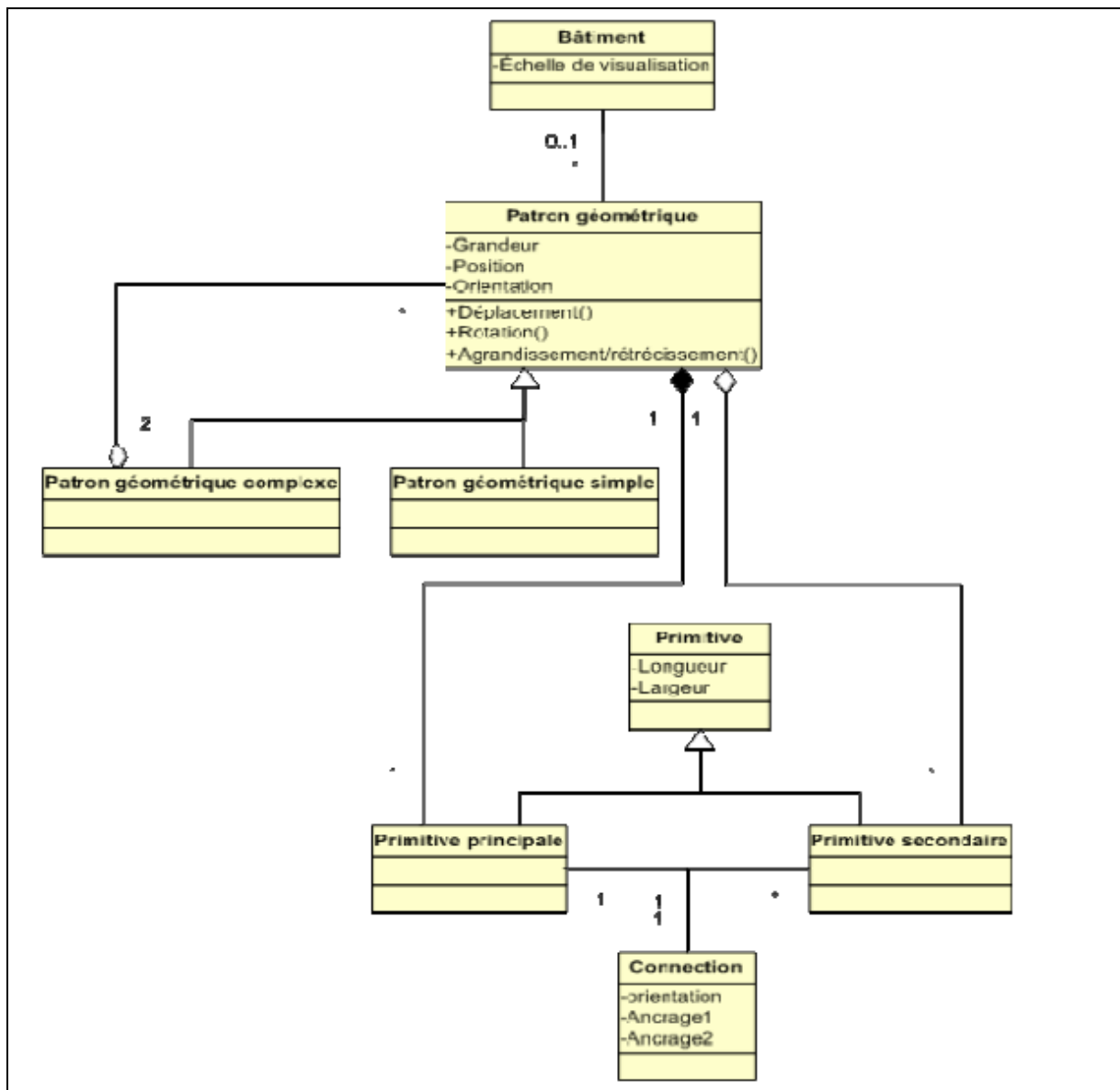


Figure 3.2 Modèle UML pour représentations multiples de Bobzien [15]

### 3.2.3 La méthode combinée

La méthode combinée [40] consiste à mémoriser des échelles pivots qui ne sont pas successives et à générer les échelles restantes d'une manière automatique à partir du bon pivot. [76] utilise une approche basée sur les patrons géométriques combinés à des algorithmes de généralisation. Les patrons géométriques sont des objets préconçus sauvegardés dans une bibliothèque (des bâtiments, des ronds-points, etc). Les objets de la carte peuvent ainsi être remplacés par les patrons et ensuite, ils peuvent être généralisés (Figure 3.3).



**Figure 3.3 Diagramme de classe UML représentant le patron géométrique [76]**

[40] combine des représentations multiples avec un algorithme génétique pour réaliser les autres échelles automatiquement afin de généraliser des routes et des bâtiments. Les algorithmes génétiques au lieu de tester toutes les solutions possibles convergent vers une solution optimale. On a aussi [22, 23, 47] qui présentent une approche qui combine la représentation multiple et la généralisation à la volée.

Cependant, toutes ces approches souffrent d'un manque en algorithmes et mesures de généralisation pour offrir un résultat qui approche le travail d'un expert en généralisation cartographique.

### **3.3 Détail de la généralisation automatique**

Les outils de la généralisation peuvent être regroupés en quatre types d'outils à savoir : le processus de généralisation, les mesures de généralisation, les algorithmes géométriques et les algorithmes de généralisation.

#### **3.3.1 Le processus de généralisation**

Le processus est le cerveau de la généralisation, il est censé remplacer le cartographe. Il peut être simplement une suite d'étapes bien définies, un modèle [75] ou un processus intelligent à base d'agents [29], de réseaux de neurones [65] ou encore d'algorithmes génétiques [95].

#### **3.3.2 Les mesures de généralisation**

Les mesures permettent d'avoir des informations sur la situation sur la carte, elles permettent de détecter les conflits, et fournissent des mesures qui permettront de choisir les bons algorithmes et d'établir leur paramétrage, elles permettent ensuite de vérifier si l'opération a bien fonctionné afin de choisir un autre traitement dans le cas contraire. Elles sont en d'autres termes les yeux du processus de généralisation.

#### **3.3.3 Les algorithmes géométriques**

Les algorithmes géométriques ne sont pas propres à la généralisation, mais ils lui sont indispensables. Nous retrouvons dans cette catégorie le diagramme de Voronoi, la triangulation de Delaunay, l'enveloppe convexe/concave, le rectangle englobant, le rectangle englobant orienté, etc.

#### **3.3.4 Les opérateurs de généralisation**

Les opérateurs sont la décomposition de la généralisation en petites opérations. Il y a plusieurs classifications des opérateurs dans la littérature, nous avons choisi celle de [65]. Ce dernier décompose les opérateurs en trois familles (Figure 3.4). Nous avons tout d'abord les

opérateurs de simplification tel que le filtrage, le lissage, la simplification de contours, la sélection. Ensuite, nous avons les opérateurs de caricature qui consistent en la dilatation, la sélection, la fusion, la translation, la typification, la schématisation, l'agrégation, le déplacement. Enfin, nous avons les opérateurs d'harmonisation comme l'équilibrage (des tailles ou des formes).

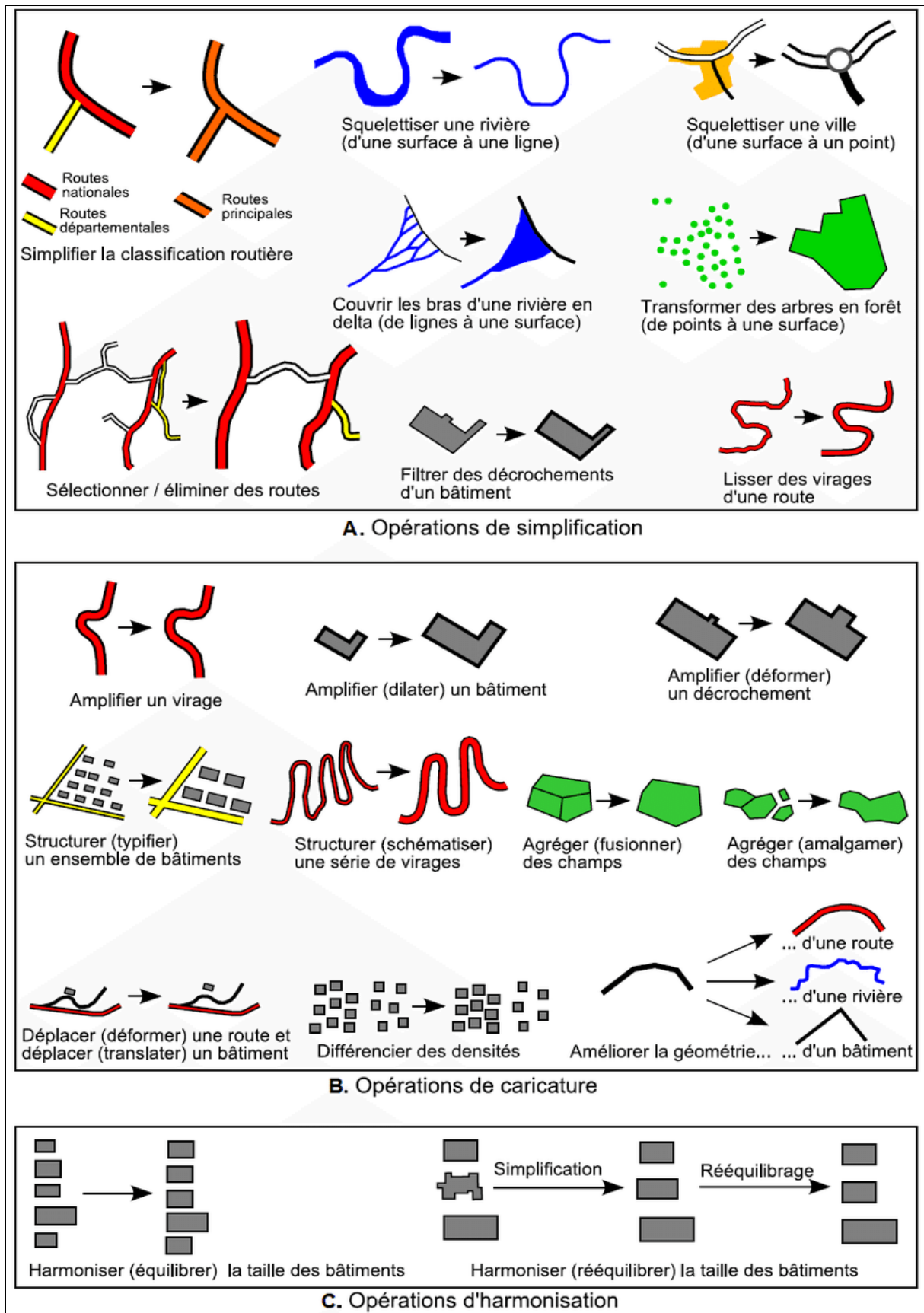


Figure 3.4 Classification des opérateurs de généralisation cartographique [65]

### 3.3.5 Les algorithmes de généralisation

Un algorithme est l'implémentation d'un opérateur de généralisation, un même opérateur peut avoir différentes stratégies pour le réaliser, la différence consiste dans l'optimisation, le temps de réponse, le nombre d'objets traités ou encore la qualité du résultat. Nous pouvons regrouper les algorithmes selon les types d'objets : Les algorithmes pour objets surfaciques [55, 73], Les algorithmes pour objets linéaires [54, 96] et les algorithmes pour objets ponctuels. Pour une liste exhaustive nous suggérons de se référer à [5, 56, 46, 60].

#### 3.3.5.1 Exemples d'algorithmes sur objets surfaciques

Nous avons par exemple l'algorithme de simplification de bâtiments de [24]. Lorsque les détails d'un bâtiment sont trop petits pour être représentés, une simplification de la forme du bâtiment peut être nécessaire (Figure 3.5). Le principe de cette simplification se base sur la recherche et la hiérarchisation des détails du contour d'un bâtiment. Ils sont ensuite traités selon leur taille (du plus petit au plus grand). Un détail peut être éliminé ou simplifié selon son importance.

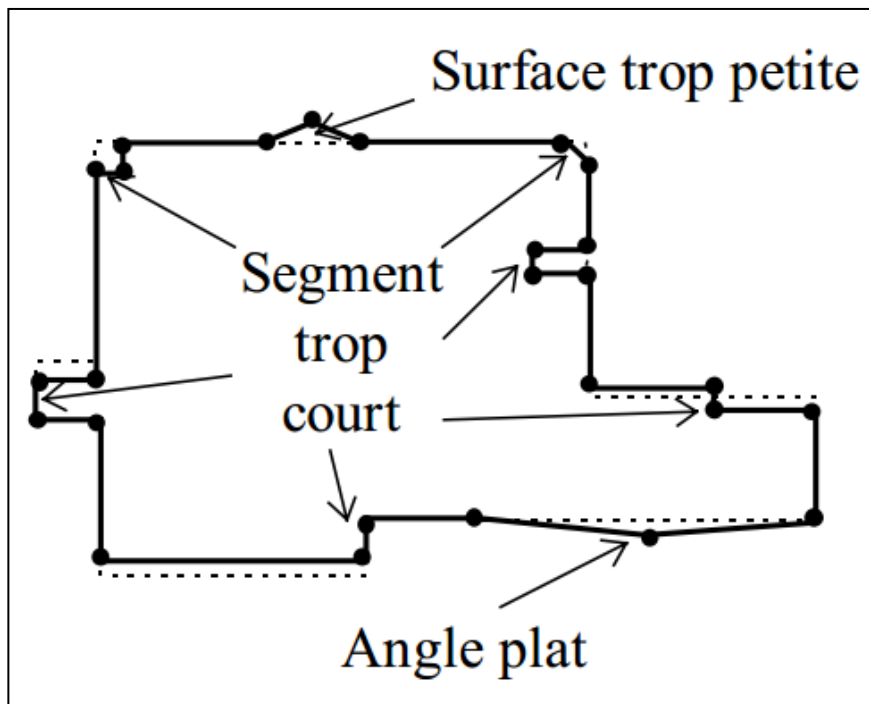
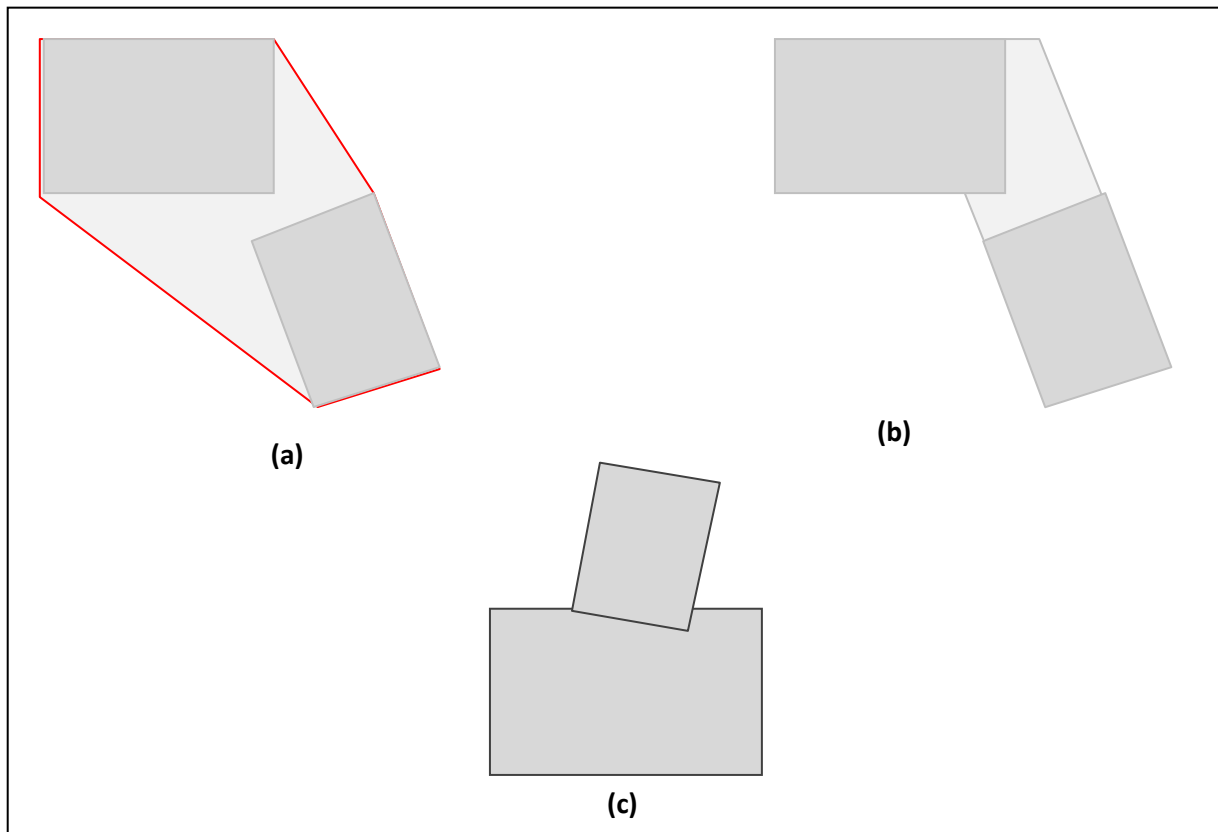


Figure 3.5 Simplification de contours du bâti [24]

Nous avons aussi l'amalgamation de Regnauld [73]. Ce dernier propose une amalgamation avec trois principes différents (Figure 3.6). Le premier consiste à remplacer deux bâtiments par leur enveloppe convexe mais cette solution propose une forme de bâtiments qui ne correspond plus à la forme des bâtiments originaux, elle est donc applicable à des bâtiments qui ont une superficie et une distribution qui s'approche de la forme de leur enveloppe convexe. Le deuxième principe consiste à étendre les segments des bâtiments et de calculer les intersections de ces derniers entre eux. Une nouvelle forme émerge en la somme des bâtiments et de l'espace engendré entre eux. Le dernier principe consiste à déplacer les bâtiments l'un vers l'autre et de construire un nouveau bâtiment par la somme des deux anciens.



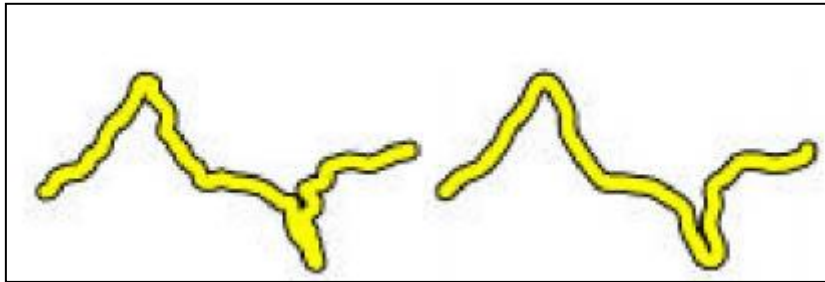
**Figure 3.6 Amalgamation par enveloppe convexe (a), amalgamation par projection (b) et amalgamation par déplacement (c) [73].**

### 3.3.5.2 Exemples d'algorithmes sur objets linéaires

Nous avons par exemple l'algorithme Plâtre (Figure 3.7) de traitement des routes [34].

L'algorithme Plâtre peut être appliqué à une route pour éliminer l'empatement et les petits

détails de cette dernière. Il consiste à transformer la route en une fonction. Cette fonction est calculée par la courbure des points de la ligne selon l'abscisse curviligne en utilisant une tangente de la ligne au point concerné. Cette fonction est lissée. Ensuite, la ligne originale est recalculée à partir de la fonction. Enfin, les extrémités et les virages sont ramenés à leur position initiale s'ils ont été déplacés.



**Figure 3.7 L'algorithme Plâtre [34]**

Nous avons aussi l'algorithme de lissage de [61]. La nouvelle position pour chaque point  $n$  de la ligne est calculée en utilisant les coordonnées des  $n-k$  et  $n+k$  points environnants. Les points voisins sont pondérés sans prendre en compte leur distance au point  $n$  mais selon leur numéro.

### **3.3.5.3 Exemples d'algorithmes sur objets ponctuels**

Nous avons par exemple la sélection. La sélection consiste à sélectionner un certain nombre d'objets pour représenter le groupe, lorsqu'il n'y a pas de place pour représenter la totalité des objets. La sélection des points à garder / des points à éliminer peut se faire selon plusieurs principes : d'une manière aléatoire, selon la densité, selon la proximité des uns aux autres, etc. Nous avons aussi la couverture. La couverture consiste à remplacer un groupe d'objets par un seul objet qui recouvre l'ensemble. La couverture peut être simplement calculée par une enveloppe convexe.

### **3.3.6 Paramétrage des algorithmes**

Un algorithme produit un résultat différent selon la valeur qu'il reçoit en paramètre. Par exemple le paramètre peut contrôler le taux de lissage d'une ligne ou la distance de déplacement d'un bâtiment. Il n'est pas toujours évident de trouver le bon paramètre pour un algorithme.

## **3.4 Synthèse**

L'analyse de la littérature, nous a permis de recenser un certain nombre de thématiques qui ne sont pas largement couvertes et ceci à cause du peu de travaux ou du nombre de variétés des formes des objets géographiques et des cas particuliers. Il est dans ce cas difficile d'avoir un seul algorithme pour tous les types d'objets. Le manque se situe plus précisément au niveau tout d'abord des séries de virages pour résoudre l'empâtement routier, c'est comme nous allons le voir à cause des variétés des types de virages et d'axes routiers. Ensuite au niveau de la détection et la généralisation des patterns de bâtiments, nous nous sommes intéressés à la typification.

## **3.5 Synthèse sur les algorithmes de traitement des séries de virages**

Il existe plusieurs algorithmes dans le répertoire de la généralisation cartographique qui ont la capacité de résoudre l'empâtement routier [2, 61, 62, 63, 77, 87, 96], il y a les algorithmes qui s'appliquent à une route entière, à savoir les algorithmes de filtrage [93, 100, 87, 92]. Ces algorithmes filtrent certains détails de la forme en éliminant des points de la ligne selon leur propre stratégie. L'algorithme de [84] ajoute des contraintes de topologie et de direction à l'algorithme de Douglas et Peuker [28] pour accélérer la simplification des polygones. L'algorithme de [81] combine des structures de données sémantiques avec des techniques de compression pour préserver les attributs des entités linéaires lors de la simplification. Il existe aussi l'algorithme de [48] qui utilise des mesures pour décider d'ignorer ou de garder des points, il prend trois points successifs et calcule les trois longueurs de segment ainsi que les angles formés par ces points qu'il déduit alors de  $\pi$ . On dispose

également de l'algorithme de [86] qui utilise une grille dont la taille de cellule est un paramètre de l'algorithme et le principe consiste à ne conserver qu'un seul point par cellule en ayant recours par exemple à une position moyenne. Enfin, nous avons l'algorithme de [28] qui est un algorithme récursif qui prend un premier segment de la droite (généralement les extrémités de la ligne) et recherche le point le plus proche, le processus est répété jusqu'à ce que la distance au segment soit en dessous du seuil, les points restants peuvent être négligés.

On a ensuite les algorithmes de lissage qui polissent certains détails de la ligne cette fois pas en supprimant des points mais en les déplaçant un peu (en utilisant par exemple une moyenne des positions des points par rapport des uns aux autres). On retrouve parmi ces algorithmes, le lissage gaussien [68] qui convole la ligne avec une fonction en forme de cloche gaussienne pour donner une forme lisse à la ligne. On retrouve aussi l'algorithme de [67] qui lisse la ligne en passant une boule le long de sa forme. Puis, on a l'algorithme de [77]. Ce dernier utilise des fonctions B-spline et des ondelettes pour obtenir le lissage.

Enfin, nous avons l'algorithme de plâtre [34], un algorithme qui donne effectivement un aspect plâtré à la route et parmi ses nombreuses utilisations, il élimine les petits virages (en les lissant) et amplifie les virages serrés.

Il existe encore des algorithmes qui s'appliquent à un virage isolé, comme MaximumBreak [65] qui consiste à dilater le contour du virage pour l'agrandir. Son principe diffère de celui de MinimumBreak qui minimise l'emprise tandis que MaximumBreak a tendance à l'agrandir. Nous avons aussi Baudruche [51] qui consiste à pousser les points du virage de tous côtés, de la même manière que l'on gonfle un ballon de baudruche. Enfin, nous avons Accordéon pour virage isolé [68] qui consiste à étendre les points de part et d'autre du virage.

Il existe des algorithmes à appliquer sur une série de virages comme accordéon pour séries [68] qui agit comme un accordéon sur la série, l'étirant des deux côtés. Le but de cet algorithme est d'étendre une série de virages successifs à la manière d'un accordéon. Son principe consiste à translater les points de la ligne des deux côtés dans le sens de la ligne directrice. La ligne directrice est obtenue par la méthode des moindres carrés. Les nouvelles coordonnées des points sont calculées par l'équation 3.1.

$$\begin{aligned} X'_i &= X_i + \varepsilon \times \text{sens} \times b \times \left( \frac{s}{s_v} + (v - 1) \right) \\ Y'_i &= Y_i + \varepsilon \times \text{sens} \times a \times \left( \frac{s}{s_v} + (v - 1) \right) \end{aligned} \tag{3.1}$$

$\varepsilon$ : est la distance de déplacement.

sens: est la direction de l'expansion, elle prend les valeurs 1 ou -1.

a, b: sont les coordonnées de la ligne directrice.

v: est le numéro du virage.

s: est la distance curviligne (le long de la ligne) entre le premier point d'inflexion du virage v et le point i.

$s_v$ : est la distance curviligne entre les deux points d'inflexion du virage v.

Cet algorithme ne convient pas à toutes les séries de virages «L'algorithme est particulièrement bien adapté au cas de série de virages dont l'axe longitudinal est rectiligne et dont les virages sont simples » [68]. Il a aussi tendance à faire déborder la série de ses extrémités «Mais cela entraîne le déplacement des deux extrémités de la série de virages, donc il est nécessaire de modifier ce décalage avec une propagation sur les deux parties de la courbe qui prolongent la série de virages. » [51].

Nous avons encore, l'algorithme Schématisation [51], ce dernier choisit des virages pour être supprimés afin de laisser de l'espace pour prolonger les virages restants. Le principe est de sélectionner un nombre pair de virages pour être supprimé en utilisant certains critères (tels que la taille, la position, l'importance, la singularité, ..), puis de supprimer ces virages ce qui aboutit à la production de deux séries de virages séparées, ces deux parties sont prolongées jusqu'à ce qu'elles se rencontrent et se connectent, on obtient alors une seule série de virage. Cet algorithme ne provoque pas de débordement des extrémités de la série, cependant, il ne convient pas aux séries de deux ou trois virages. Il ne s'applique qu'aux séries dont l'axe est droit et les virages simples comme Accordéon.

### 3.5.1 Analyse

Les algorithmes traitant des séries de virages ne sont pas nombreux. De plus, chaque algorithme a ses points forts et ses points faibles. Accordéon et Schématisation ne permettent de généraliser que certains types de virages et certaines séries, mais encore le premier augmente l'emprise terrain et le second supprime des virages.

## 3.6 Synthèse sur les mesures de détection des patterns de bâtiments

Dans la littérature, nous trouvons les algorithmes de clustering. Ils permettent de construire des groupes de bâtiments basés sur la proximité [98]. [21] présente quatre algorithmes représentant les algorithmes existants et propose une comparaison entre eux. Ces algorithmes sont le Minimum Spanning Tree (MST) [30, 31], le DBSCAN (Density-Based Spatial Clustering) [32], le CHAMELEON [49] et un ASCDT (Adaptive Spatial Clustering based on Delaunay Triangulation) [26]. L'évaluation est basée sur les caractéristiques de distribution des groupes en utilisant deux méthodes, le S\_Dbw [21] et les cercles d'évaluation des clusters. L'expérience montre que DBSCAN et ASCDT sont supérieurs à CHAMELEON et MST. Résultats confirmés par [27]. Ces algorithmes ne conviennent pas à notre cas. Avec ces types de méthodes, il est difficile d'isoler un motif du reste des bâtiments.

Il existe également des méthodes de détection automatique de modèle bien précis [59] comme la détection de modèle linéaire et la détection de modèle en grille [69, 70, 31]. [99] a permis de détecter le motif colinéaire, le motif curviligne, les bâtiments le long de la route et le motif de la grille. [91] détecte à peu près les mêmes modèles.

Il existe également des méthodes de détection de pattern à partir de modèles. [97] propose de détecter des motifs en les comparant à un modèle enregistré en mémoire. Chaque modèle est décrit par un ensemble de mesures telles que la distance, la différence d'orientation, le rapport de taille, la différence d'allongement et le rapport de parement. Il est toujours possible d'étendre la bibliothèque de modèles pour identifier plus de modèles, mais il n'est pas possible d'identifier un modèle qui ne se trouve pas dans la bibliothèque.

### **3.6.1 Analyse**

Les algorithmes de clustering ne permettent pas d'isoler un motif du reste des bâtiments. Les algorithmes de détection de modèle ne permettent pas d'identifier un motif qui ne se trouve pas en mémoire. Il n'y a pas de travaux pour détecter les motifs répétés (comme le motif d'un tissu). Et il y a peu de travaux qui proposent des algorithmes de généralisation pour traiter les patterns de bâtiments.

## **3.7 Synthèse sur les algorithmes de typification de bâtiments**

On peut regrouper les travaux en typification de bâtiments selon le type de pattern, nous les divisons en trois catégories :

### **3.7.1 Typification de groupes structurés en pattern linéaire**

[45] propose une typification en plusieurs étapes : Tout d'abord calculer le nombre  $m$  (équation 3.2) de bâtiments à sélectionner en utilisant  $n$ , le nombre de bâtiments à l'origine et  $f$ , le facteur de changement de taille des composants.

$$m = \frac{n}{f} \quad (3.2)$$

En second lieu, calculer les  $m$  positions des  $m$  futurs bâtiments en utilisant la formule suivante :  $m$  représente les positions ( $c_{r,i}$ ) qui sont calculées en utilisant les positions originales ( $c_{o,P(i)}$ ) (Ensemble d'équations 3.3).

$$c_{r,i} = c_{o,P(i)} + \lambda(i) \times c_{o,P(i)} c_{o,Q(i)}$$

$$R = (i - 1) \times \frac{n - 1}{m - 1} \tag{3.3}$$

$$P(i) = 1 + E(R), E(R) \text{ est le plus grand entier } \leq R$$

$$\lambda(i) = R - E(R)$$

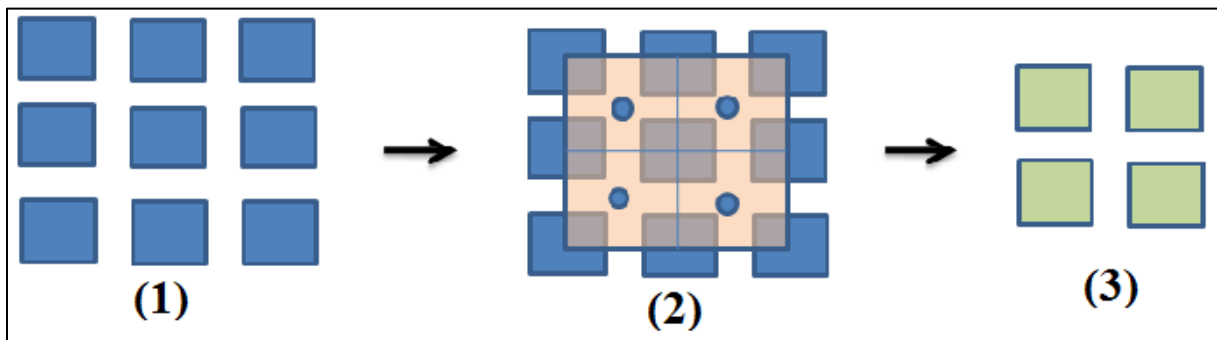
$$\text{Si } \lambda(i) = 0 \text{ alors } Q(i) = P(i), \text{ sinon } Q(i) = P(i) + 1$$

Ensuite les bâtiments se trouvant le plus près de ces emplacements (positions) sont sélectionnés avec une préférence pour les bâtiments remarquables. Enfin les bâtiments sont simplifiés ou remplacés par des rectangles.

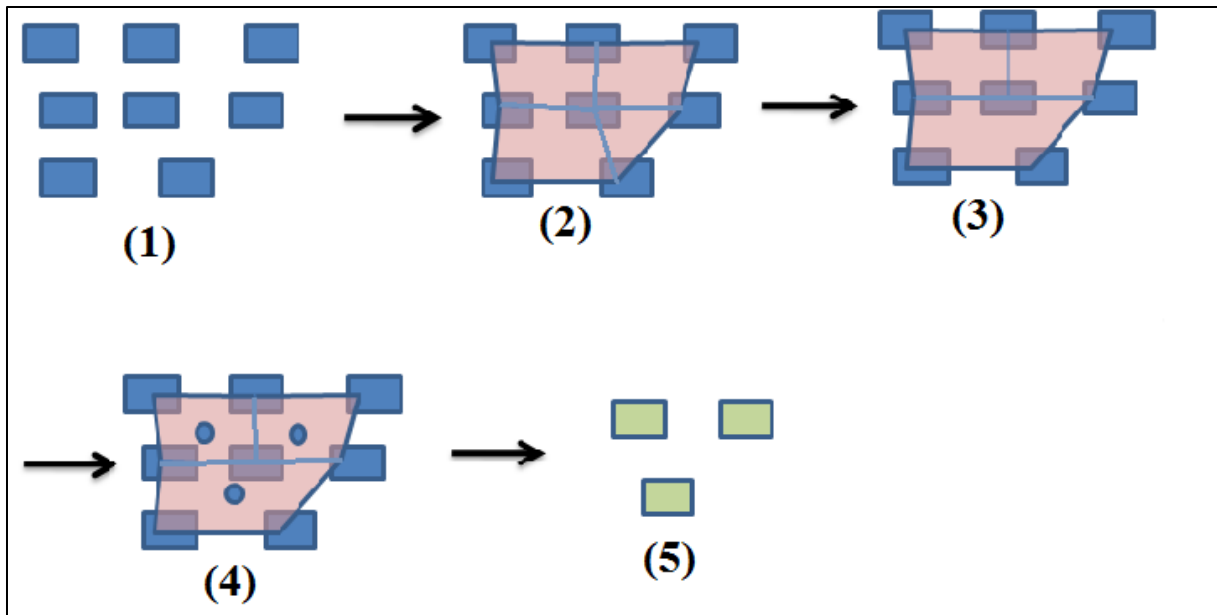
Il y a les méthodes qui considèrent la typification comme un processus progressif et itératif consistant en l'élimination, l'exagération et le déplacement. [38, 12] utilise la méthode du *Minimum Spaning Tree* (Arbre de Recouvrement Minimum) pour le regroupement des bâtiments. Pour [72] la typification est aussi un processus itératif et progressif, sa méthode consiste en ceci : Les bâtiments sont tous qualifiés selon des mesures de la taille, la forme, l'orientation, l'importance. Ensuite, ils sont placés tour à tour autour de leur centre de gravité jusqu'à ce qu'il n'y ait plus de places. Durant ce processus, les bâtiments sont déplacés d'une manière homogène, ils sont simplifiés, érodés, remplacés par un rectangle ou amalgamés, selon la situation. Cependant cet algorithme se substitue quelque peu au processus de généralisation.

### 3.7.2 Typification de groupes structurés en pattern grille

Pour la typification de grilles, nous avons [12] qui propose une méthode itérative. [6] propose une approche basée sur les moindres carrés. [89] proposent une typification de grilles absolues (Figure 3.8) et de quasi-grilles (Figure 3.9) par une méthode en trois étapes : calcul d'un graphe de voisinage par une triangulation de Delaunay des sommets des bâtiments (contour), fusion des triangles de Delaunay pour former des cellules agrégées, positionnement des nouveaux bâtiments (représentés par des rectangles englobant orientés) aux centres de ces cellules.



**Figure 3.8 Typification de grille parfaite. En (1) nous avons la grille originale. En (2) les triangles des bâtiments adjacents ont été fusionnés formant des polygones. En (3) les nouveaux bâtiments ont été placés aux centres des polygones de l'étape précédente. [89]**



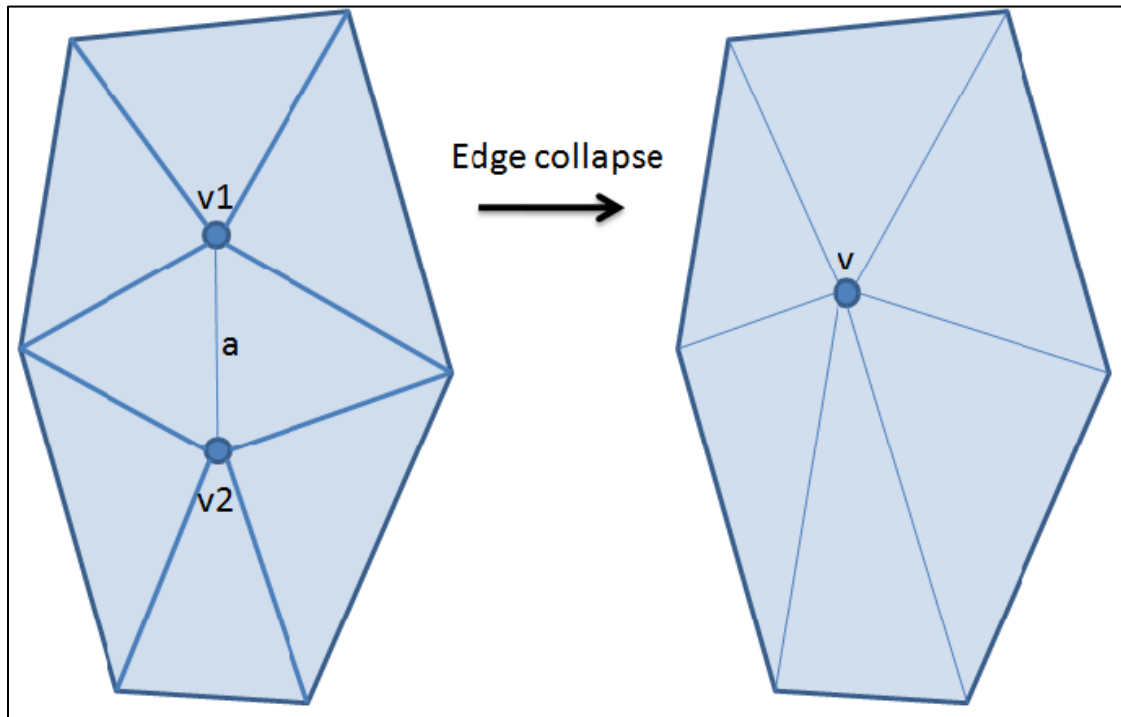
**Figure 3.9 Typification de quasi-grille. En (1) nous avons la grille originale. En (2) les triangles des bâtiments adjacents ont été fusionnés formant des polygones. En (3) les polygones ont été fusionnés aussi. En (4) les nouveaux bâtiments ont été placés aux centres des polygones de l'étape précédente. En (5) nous avons le résultat de la typification [89]**

### 3.7.3 Typification du groupe non structuré

Pour les motifs non structurés, [53] proposent une typification par trois étapes: calcul du nombre de bâtiments du groupe à l'échelle cible en utilisant une loi radicale améliorée, suppression itérative des bâtiments du groupe, ensuite une harmonisation des tailles des bâtiments restants.

[17] proposent une méthode itérative en plusieurs étapes : calcul d'un diagramme de Delaunay des centres des bâtiments, recherche du segment de Delaunay le plus petit et suppression des deux sommets qui le définissent, ils sont remplacés par un seul sommet au milieu des deux ( Figure 3.10). Les bâtiments sont ensuite placés et remplacés par des rectangles. [17] proposent d'ajouter un facteur de densité  $f$  aux groupes de bâtiments pour ne pas déséquilibrer la différence de densité entre groupes lors de la suppression de bâtiments. Le

problème avec cette méthode c'est qu'elle se base sur la distance entre centre de bâtiments qui peut ne pas refléter la véritable distance entre deux bâtiments (cas d'un grand bâtiment avec un petit bâtiment). [91, 6] Proposent une méthode similaire.



**Figure 3.10 Fusion (unification) des sommets de Delaunay [17]**

#### 3.7.4 Analyse

Il existe une grande variété de patterns de bâtiments, mais encore il y a même des patterns disposés en pattern (pattern de patterns). Chaque pattern nécessite une détection et une typification propre. Les types de patterns ne sont pas tous pris en charge par les algorithmes de typification existants.

### 3.8 Conclusion

Nous avons présenté dans ce chapitre un aperçu sur le domaine de la généralisation cartographique. Nous avons analysé les travaux dans le domaine et mis en évidence certains manques, dont le traitement des séries de virages et les patterns de bâtiments. Dans les prochains chapitres nous répondons à ce manque avec la proposition de nouveaux outils de généralisation.

## Chapitre 4 : Proposition d'un nouvel algorithme pour les séries de virages (MBRS)

Dans ce chapitre, nous présentons notre contribution dans le domaine du routier. Nous y exposons en détail notre algorithme de généralisation des séries de virages pour corriger l'empâtement nommé MBRS (Minimum-Break for Road Séries)

#### 4.1 Introduction

Dans le chapitre précédent, nous avons pu mettre en relief certains besoins du domaine de la généralisation automatique. Dans ce chapitre et les chapitres suivants nous répondons à ce manque en proposant trois nouveaux outils de généralisation. Le tout premier étant le sujet de ce chapitre l'outil MBRS (Minimum-Break for Road Series [43]) un nouvel algorithme pour les séries de virages. Notre algorithme est basé sur Faille-Min, un algorithme pour virage isolé que nous avons adapté aux séries de virages mais avant, il est nécessaire de voir certaines notions du domaine.

#### 4.2 Qu'est-ce que l'empâtement?

L'empâtement se produit lorsque le symbole de la route se superpose à lui-même, rendant la route illisible et produisant un effet visuel inesthétique comme le montre la figure 4.1.

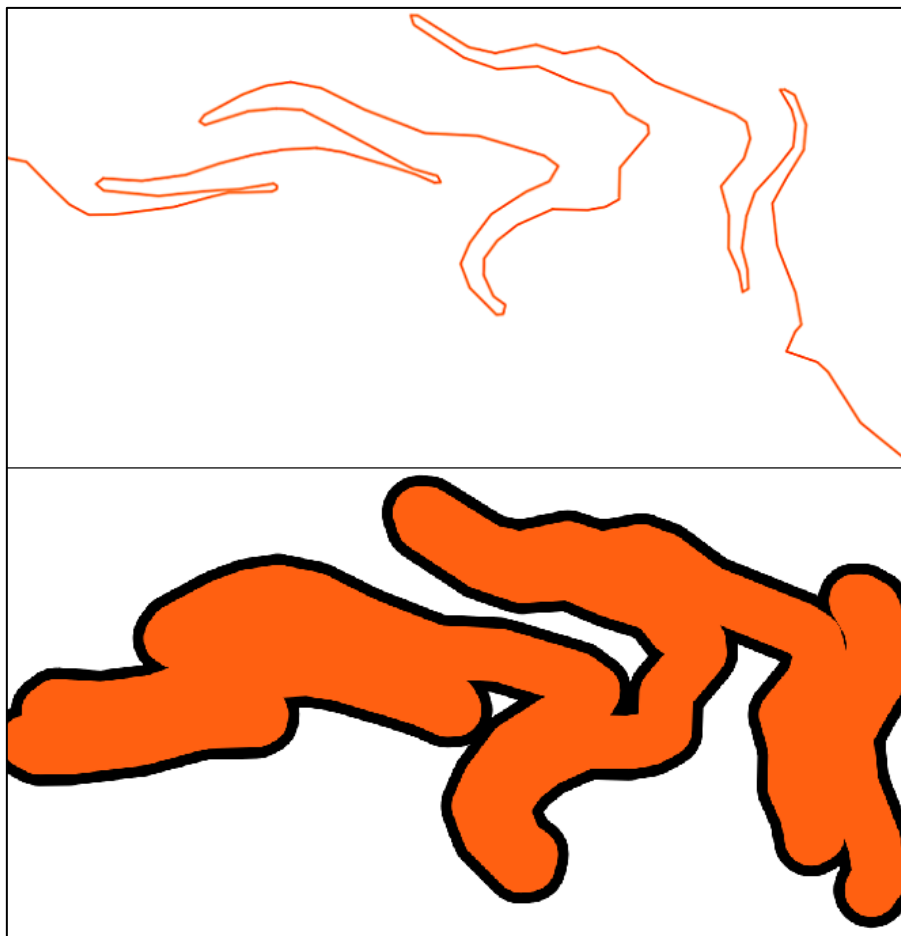


Figure 4.1 Les effets de l'empâtement

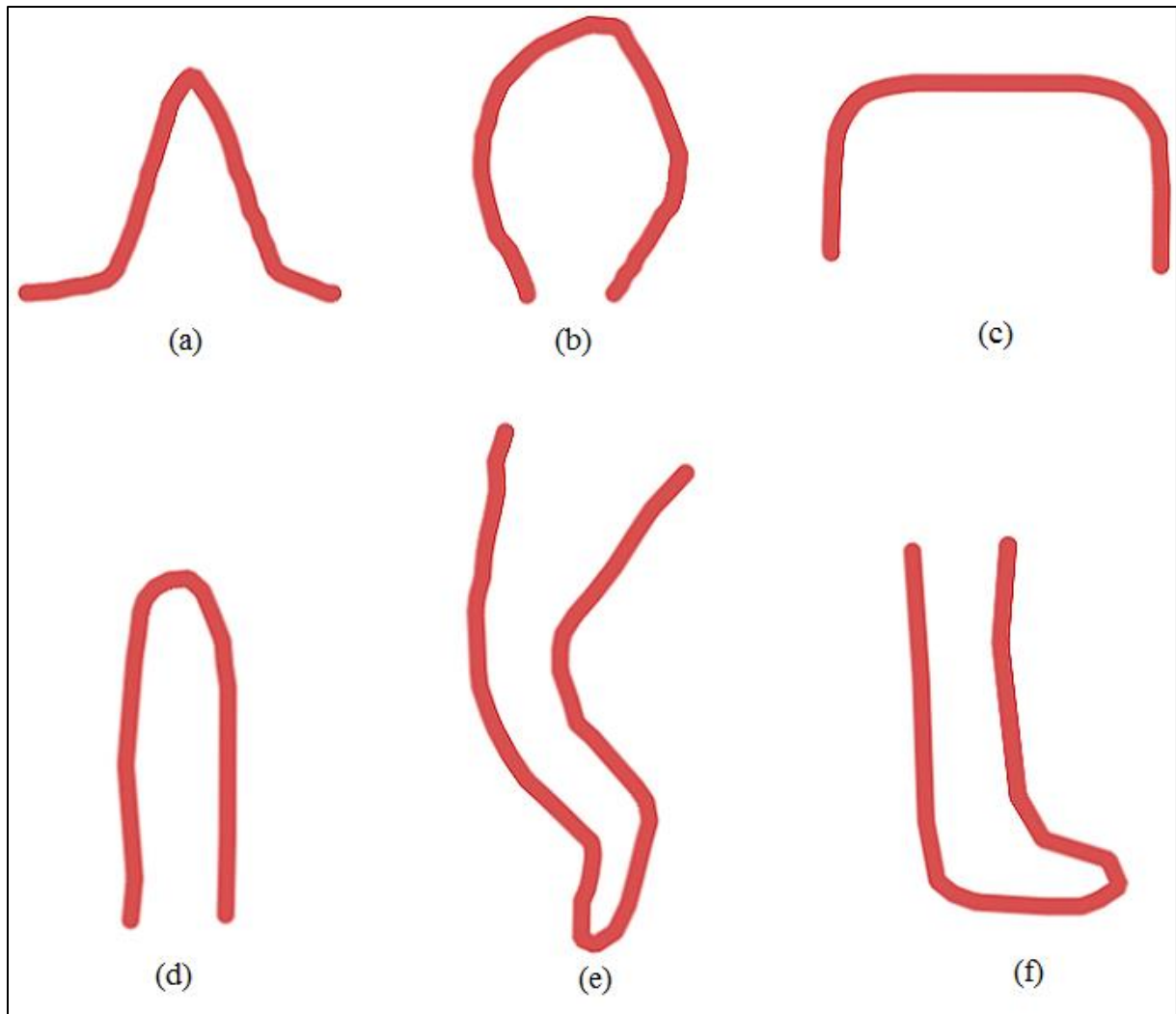
[65] a défini l'empâtement autrement: « La ligne symbolisée est considérée comme empâtée si et seulement si la distance entre l'un de ces bords et son axe est supérieure à ou égal à 1,7 fois la demi-largeur de son symbole, ou que l'un de ses bords est discontinue. Quand la ligne est considérée empâtée, si l'écart bord-axe est supérieur au seuil d'un seul bord ou si un seul bord est discontinu, l'empâtement est dit "d'un côté", sinon on dit qu'il est «des deux côtés». La valeur 1,7 est une moyenne expérimentale valeur en fonction de la lisibilité et de l'esthétique qui peut être modifiée au besoin. » Cette définition vient de l'effet que le symbole d'une route est un buffer de la ligne d'origine, et plus le buffer est grand, plus les parties de la route fusionnent. [65] agrège ensuite les parties empâtées adjacentes pour créer des zones d'empâtements: «La série parties empâtées adjacentes, ou peu séparées, sont fusionnées (nous utilisons également un seuil égale à 1,7 demi largeur du symbole). »

### **4.3 Qu'est-ce qu'un virage?**

Même si visuellement il semble facile de définir un virage, techniquement ça ne l'est pas. La littérature suggère différentes manières de le concevoir; selon certains, un virage est une zone comprise entre deux points d'inflexion [68], une zone de forte courbure [34], une zone affectée par l'empâtement [65] car les virages en général sont plus sujets à l'empâtement que des zones linéaires. Le choix de la définition peut être porté selon l'utilisation que l'on veut en faire.

### **4.4 Les formes de virages**

Après avoir étudié les routes en profondeur, [68] a défini six formes différentes de virages auxquelles d'autres peuvent être ajouté plus tard (si nécessaire). La figure Fig.4.2 montre ces différents types de virages. Pour notre part, nos échantillons s'inscrivent dans ces six formes.



**Figure 4.2 Formes de virages [67] : coude (a), boucle (b), carré (c), épingle à cheveux (d), bras (e) et chaussette (f)**

#### **4.5 Comment découper une route?**

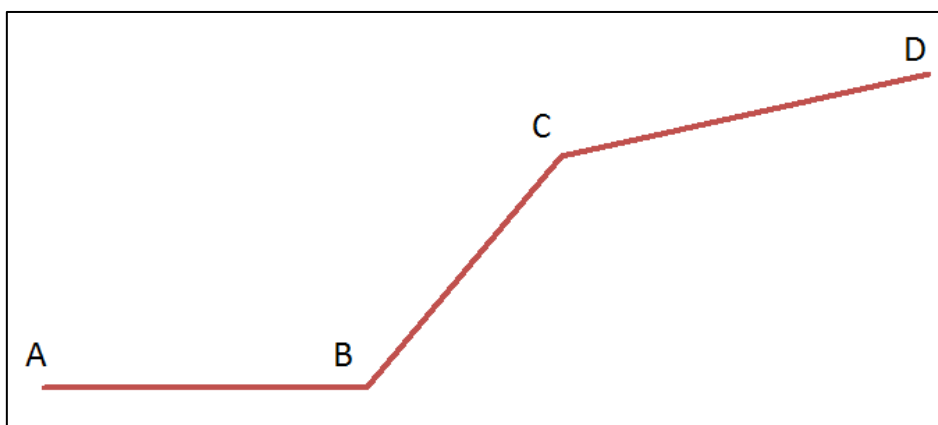
S'il existe des algorithmes qui s'appliquent à une route entière [48, 50, 51, 80, 34, 81], il existe également des algorithmes conçus pour des portions particulières d'une route [51, 64, 34], il est donc intéressant de décomposer une route pour mieux traiter ses différentes parties. Une route est une courbe qui présente visuellement différentes zones, des zones avec un virage isolé, des zones avec des séries de virages et des zones linéaires.

Dans la littérature, il y a différentes manières de diviser une route. Une division selon la mesure de la sinuosité [68, 11] divisant la route en zones non-sinueuses, peu sinueuses ou

très sinueuses. Les travaux de [54] reposent également sur la sinuosité et l'utilisation pour la mesurer de l'algorithme de [28] et celui de [83]. On peut également utiliser la mesure de la fractalité [68], l'homogénéité [77], ou d'autres mesures. Il est également possible de découper une route selon les points d'inflexion [51, 68]. [65] propose une division selon une mesure d'empâtement et divise la route en zones empâtées (d'un seul côté ou des deux côtés) et en zones non empâtées. [9] fait une classification basée sur un réseau de neurones fonctionnant avec les mesures de sinuosité et de directionnalité [58, 10]. [8] fondent leur travail sur les ondelettes.

Dans notre cas, nous cherchons à isoler les séries de virages; nous avons choisi la recherche de points d'inflexion, et nous avons choisi la méthode [67] couplée avec l'algorithme de Douglas et Peuker [26] et le lissage gaussien [67] pour éviter la sur-détection. Cette méthode est basée sur le produit vectoriel entre chaque trois segments successifs, si le signe du produit est différent entre le produit du premier segment et du deuxième et le produit du deuxième segment et du troisième, alors le milieu du deuxième segment est désigné comme un point d'inflexion. L'équation 4.1 et la figure 4.3 montrent le calcul pour la détection du point d'inflexion.

$$\overrightarrow{AB} \wedge \overrightarrow{BC} \times \overrightarrow{BC} \wedge \overrightarrow{CD} < 0 \quad (4.1)$$

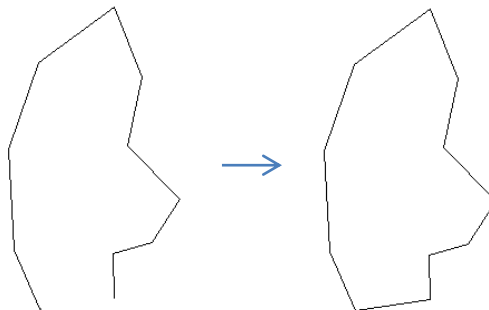


**Figure 4.3 Détection des points d'inflexion. Le milieu de [BC] sera désigné comme étant un point d'inflexion.**

#### 4.6 Algorithme Faille-Min pour les virages isolés

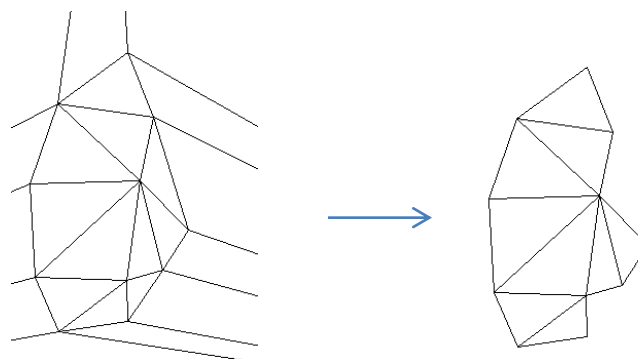
Cet algorithme utilise une astuce de cartographe qui consiste à replier le bord du virage sur lui-même pour former une ligne comme l'explique [65]. Pour imiter cette astuce dans le traitement informatique, l'algorithme préfère extraire le squelette du virage, puis dilater cette ligne squelette pour obtenir le nouveau virage. Les détails du principe sont les suivants:

- La ligne constituant le virage est fermée par un segment pour en faire une surface (figure 4.4).



**Figure 4.4 Transformation d'un virage en une surface.**

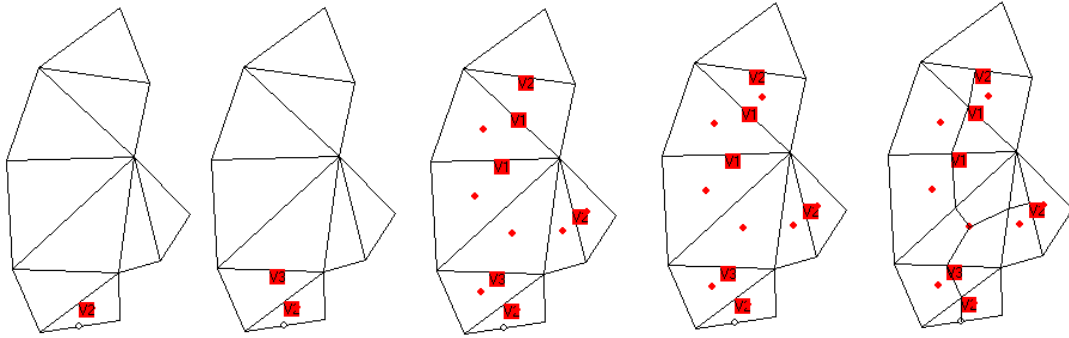
- Une triangulation de Delaunay est appliquée sur cette surface.
- Les triangles qui sont à l'extérieur de la surface sont éliminés (figure 4.5).



**Figure 4.5 Application d'une triangulation de Delaunay sur le virage.**

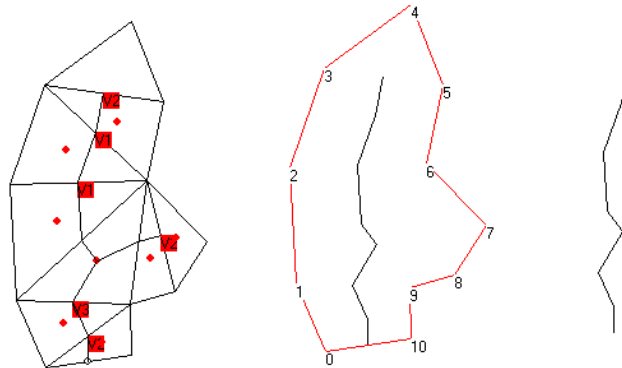
- Un arbre squelette de la surface est construit (figure 4.6) de la manière suivante:

- Si un triangle a 2 voisins, le milieu des côtés communs est connecté.
- Si un triangle a 3 voisins, le centre de gravité du triangle est relié au milieu de ses trois côtés.



**Figure 4.6** Extraction de l'arbre squelette du virage.

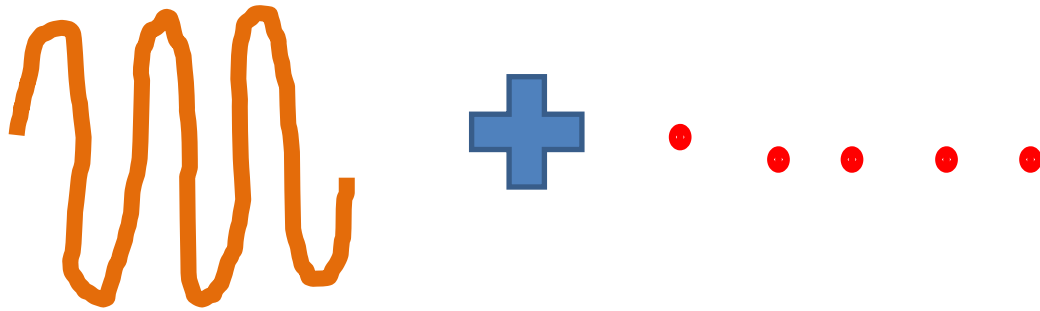
- La ligne la plus longue de l'arbre est choisie et lissée légèrement (figure 4.7).
- À la fin, la ligne est dilatée pour obtenir le virage final.



**Figure 4.7** Choix du chemin le plus long comme squelette pour le virage.

#### 4.7 MBRS (Minimum-Break for Road Series)

L'algorithme reçoit en entrée la série de virages (sous forme d'une polygone) et un tableau (vecteur) de  $n$  points d'inflexion pour délimiter les virages (figure 4.8).



**Figure 4.8** Les entrées de l'algorithme MBRS.

Deux points d'inflexion consécutifs représentent le début et la fin d'un virage. En sortie, nous avons une ligne généralisée. Les principales étapes de l'algorithme sont les suivantes:

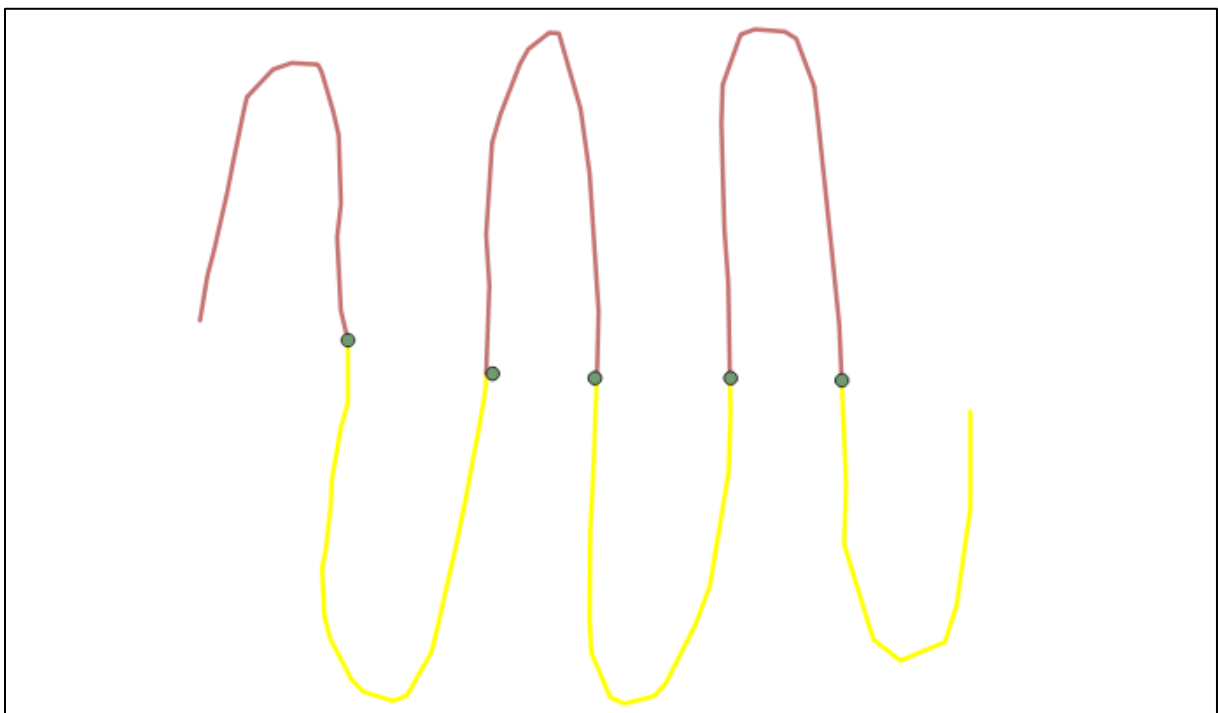
- Nous divisons d'abord la ligne en fonction des points d'inflexion en  $n-1$  virages séparés. Un virage est un sous-ensemble de points successifs dans la ligne qui commence par un point d'inflexion et se termine par le point d'inflexion suivant.
- Nous extrayons ensuite la ligne squelette de chaque virage.
- La ligne squelette est éventuellement tassée par le haut ou allongée, pour s'adapter à la hauteur réelle du virage si elle a changé à cause des étapes précédentes.
- La ligne du squelette est allongée par le bas pour faciliter l'étape de coupe.
- Nous appliquons un buffer avec un paramètre spécifique à chaque virage, qui sera détaillé dans la section suivante.
- La forme obtenue est tronquée en bas à l'aide d'une ligne supportant le segment reliant les deux points d'inflexion qui entourent le virage. Cette étape est plus facile car la ligne squelette a été allongée par le bas.
- Nous changeons les surfaces obtenues en lignes pour avoir les virages finaux. À ce stade, nous pouvons avoir des virages qui se déconnectent ou s'intersectent.
- Nous assemblons les virages en une nouvelle ligne.
- Nous appliquons des traitements de lissage et d'esthétiques à la nouvelle ligne.

- Nous calibrons la nouvelle ligne autour des extrémités de la ligne d'origine.
- Nous réintégrons la ligne dans la carte. La ligne peut déborder sur les côtés à cause d'une légère augmentation de la taille. La propagation des résultats peut être nécessaire [34, 52]. Nous pouvons le faire par un simple déplacement des lignes autour ou même en les tassant.

Les étapes détaillées de l'algorithme sont décrites ci-dessous.

#### 4.7.1 Extraction des virages

Premièrement, nous divisons la série de virages en virages individuels. Pour cette étape, nous affectons chaque point d'inflexion au segment le plus proche de la ligne constituant la série de virages. Ensuite, nous calculons sa projection sur ce segment. Nous ordonnons les points d'inflexion en fonction du numéro de segment. Ensuite, nous prenons la série de points entre chaque deux points d'inflexion pour créer nos virages (figure 4.9).



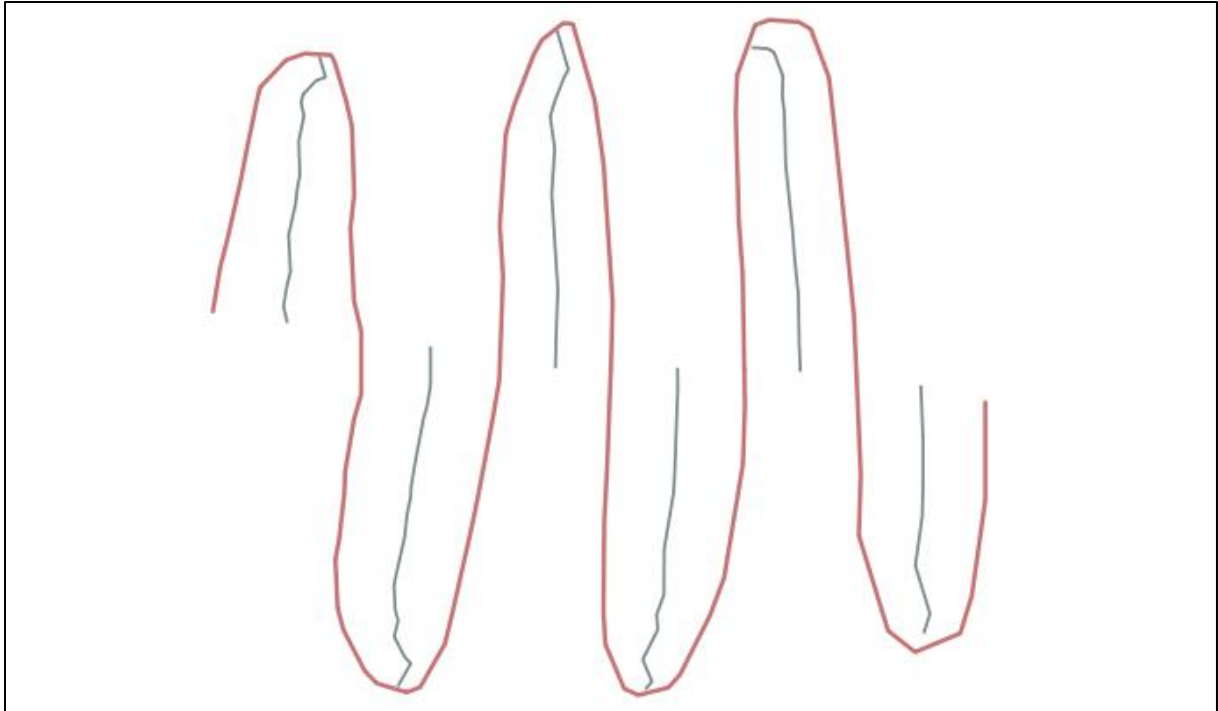
**Figure 4.9** Extraction des virages

#### 4.7.2 Extraction du squelette

Nous calculons les squelettes des virages découpés (figure 4.10) comme suit:

Pour chaque triangle:

- Nous calculons la triangulation de Delaunay
- Nous cherchons le triangle de départ (celui qui contient les deux extrémités du virage).
- On crée une structure de chemin qui contiendra tous les chemins possibles, cette structure a cinq attributs, dans l'ordre: le triangle courant, le triangle précédent (pour ne pas rebrousser chemin), un champ booléen pour indiquer que le chemin est fermé (ie nous avons parcouru tous les triangles dans cette direction), un champ booléen pour indiquer si le chemin est supprimé (un chemin est supprimé s'il bifurque dans deux directions différentes, il sera alors remplacé par deux chemins qui eux continueront), un champ contenant la liste des points constituant le chemin.
- À chaque itération, nous avançons d'une étape dans tous les chemins qui ont été créés:
  - Si le triangle courant a deux voisins, on continue dans la direction du triangle qui n'est pas le triangle précédent et on ajoute le point du milieu du segment commun aux deux triangles à la liste des points du chemin.
  - Si un triangle a trois voisins, il est supprimé et deux nouveaux chemins sont créés, ils récupéreront ses points et continueront en direction des deux voisins qui ne sont pas le précédent, sur le même principe que ci-dessus
- Nous nous arrêtons lorsque tous les chemins sont fermés ou supprimés. Nous calculons le plus long chemin de notre liste de chemins pour être notre squelette.



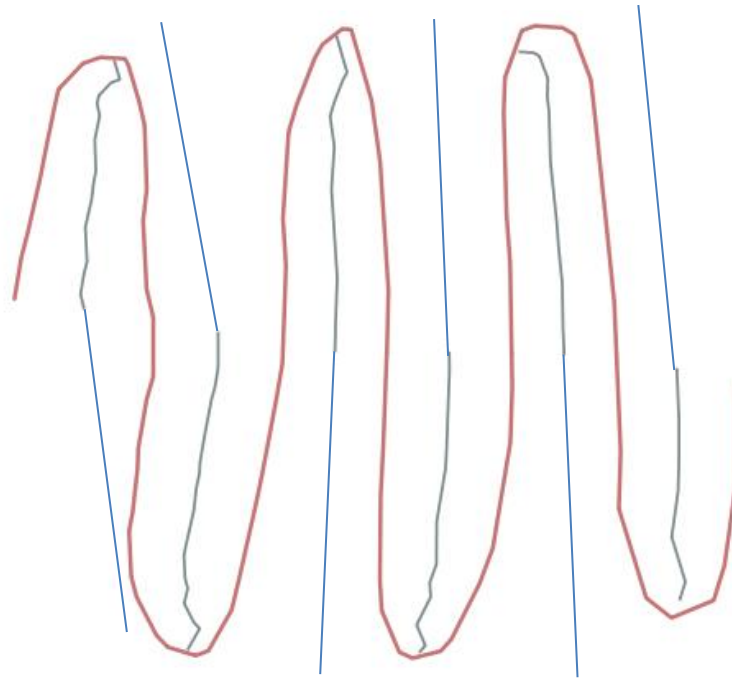
**Figure 4.10** Extraction des squelettes

#### 4.7.3 Extension des squelettes

La base du virage est étendue (figure 4.11), cela correspond à l'extension du segment numéro un du squelette. La valeur d'expansion doit être très grande, nous avons choisi une valeur aléatoire. Soit  $s(0)$ ,  $s(1)$  les deux premiers points du squelette  $s$  et  $ex$  la valeur d'extension, le point d'extension  $p$  est donné par l'équation 4.2.

$$\overrightarrow{s(1)p} = ex \times \overrightarrow{s(1)s(0)} \quad (4.2)$$

Ensuite, nous remplaçons simplement  $s(0)$  par  $p$  dans le squelette.



**Figure 4.11 Extension des squelettes.**

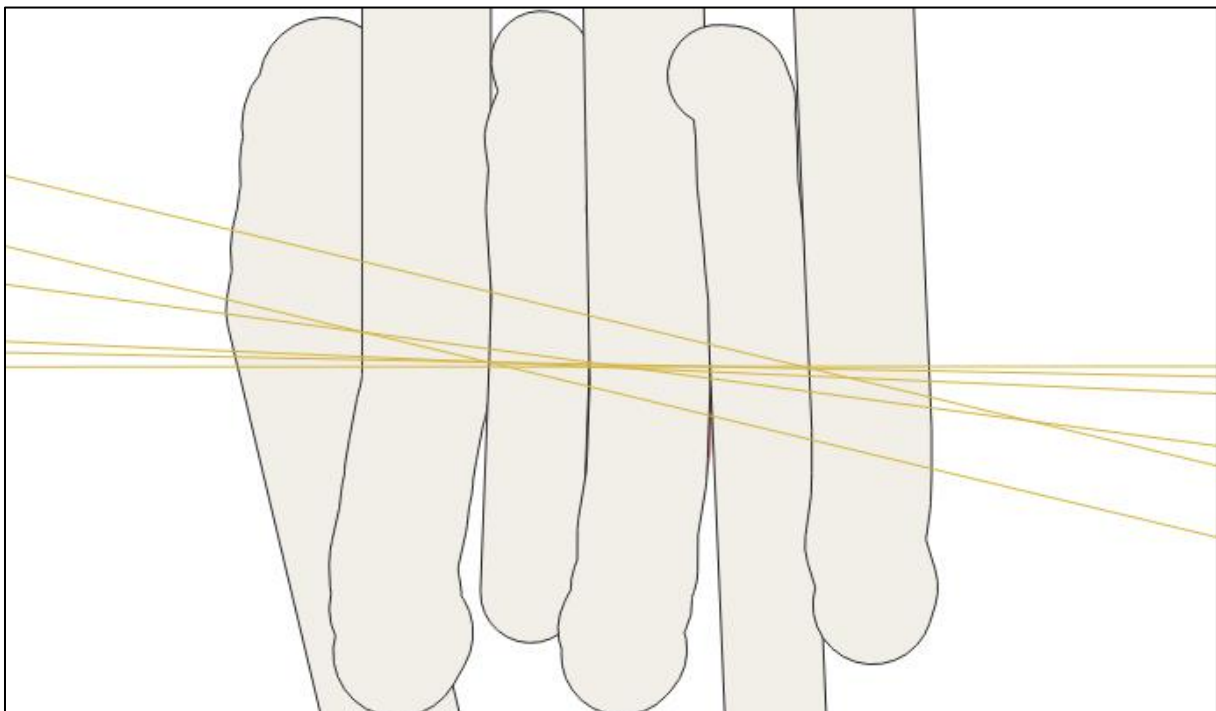
#### **4.7.4 Bufferisation**

Nous calculons ensuite les buffers des squelettes étendus (figure 4.12). Mais d'abord, nous calculons la valeur paramètres des buffers. Le paramètre pour un virage est égal à la demi-largeur du virage d'origine (donnée par la distance entre les deux extrémités de la ligne du virage). Si cette valeur est inférieure à la moitié de la valeur du symbole ajoutée au seuil de visibilité 0,2 mm (35, 85) alors la valeur du paramètre sera cette valeur. Comme cette dernière valeur peut changer la taille du virage d'origine, une harmonisation des tailles des virages est nécessaire. L'harmonisation consiste à multiplier les paramètres des autres virages par le facteur d'augmentation du virage. Le paramètre d'harmonisation est le plus grand facteur d'augmentation parmi les facteurs de virage.

Nous avons testé une autre méthode pour remplacer l'étape de buferisation. Un virage peut avoir différentes largeurs de la base à la tête. Un buffer attribue la même largeur sur toute la hauteur du virage. Cette méthode consiste à segmenter la ligne squelette en suivant un pas

régulier et à balayer au travers par une ligne perpendiculaire au squelette du virage. Il est entendu que nous devons prendre en compte les lignes de part et d'autre du squelette à la recherche des différentes largeurs du virage. Chaque largeur est affectée au point du squelette correspondant, le nombre de segments est un paramètre de l'algorithme. Ensuite, il ne reste plus qu'à augmenter les largeurs pour qu'elles soient égales à la largeur du symbole plus le seuil de visibilité, on harmonise ensuite les tailles.

Ensuite, nous construisons un nouveau virage en projetant des points de part et d'autre du squelette avec les paramètres calculés. A la fin, un demi-cercle (série de points en cercle) dont le centre est le dernier point du squelette et le rayon égal au dernier paramètre est ajouté à la tête des virages. Il est également possible d'utiliser différentes valeurs de buffer le long de la ligne ou une autre opération pour remplacer le buffer.

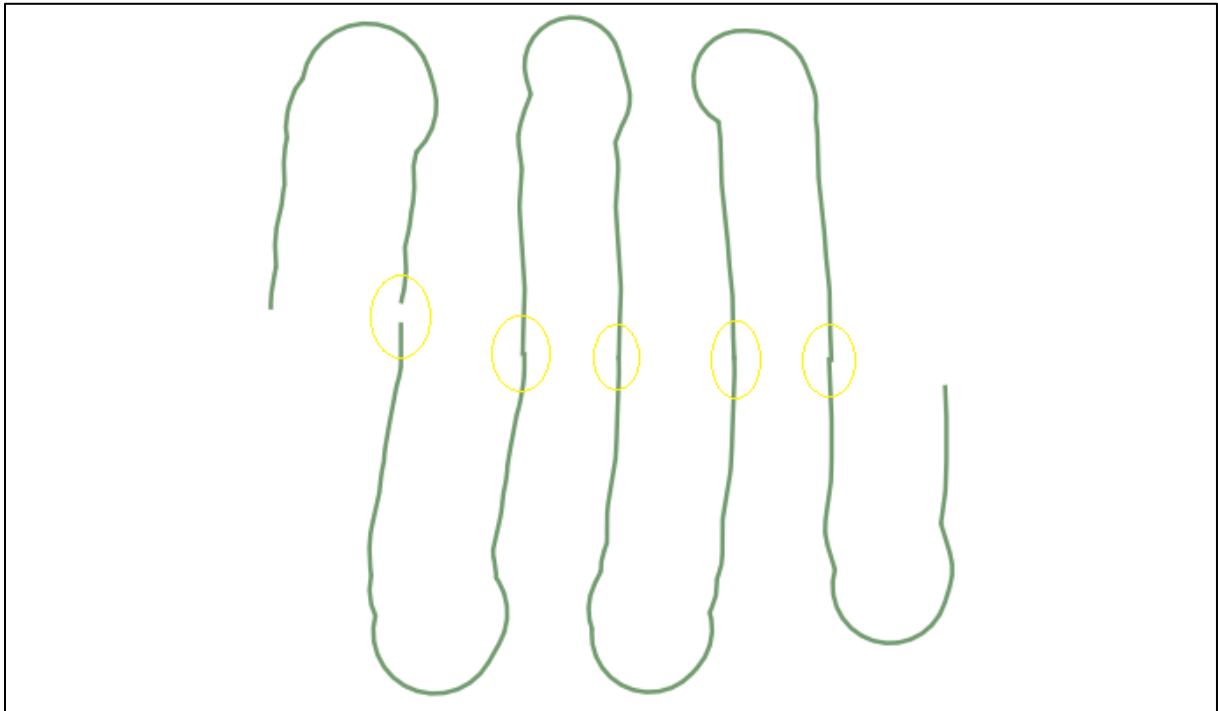


**Figure 4.12 Bufferisation**

#### **4.7.5 Découpe des buffers et extraction des virages résultants**

Cette étape consiste à extraire le virage résultant à partir du buffer (figure 4.13). Tout d'abord, nous calculons l'intersection du buffer avec la ligne portant les deux points

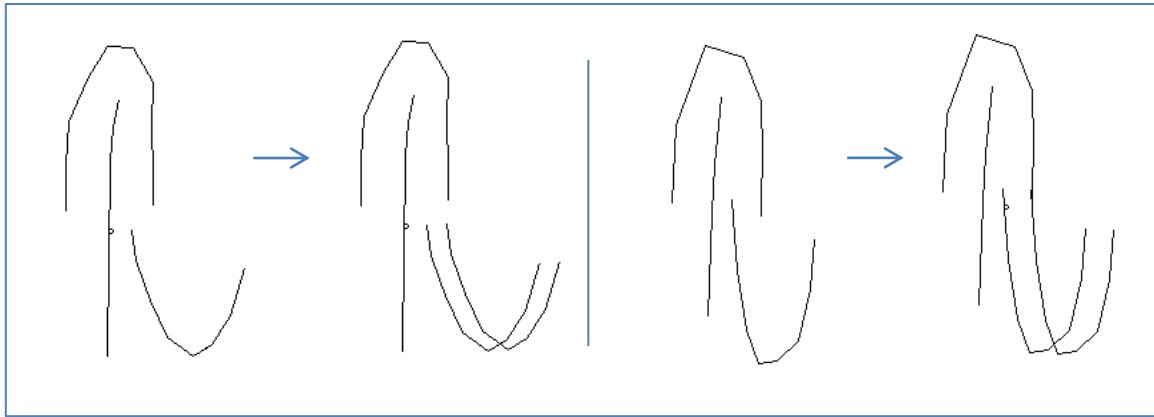
d'inflexion délimitant le virage. Ensuite, nous découpons le buffer pour obtenir deux polygones. Nous ne conserverons que le polygone dont nous avons besoin grâce à un test (ex : la polyligne du squelette étant composée de points ordonnés, on garde la partie qui contient le deuxième point du squelette, le premier point faisant partie de la partie étendue dans l'étape précédente). Ensuite, le polygone est converti en polyligne.



**Figure 4.13 Virages résultants. On remarque les problèmes de jonction (entourés par des cercles jaune).**

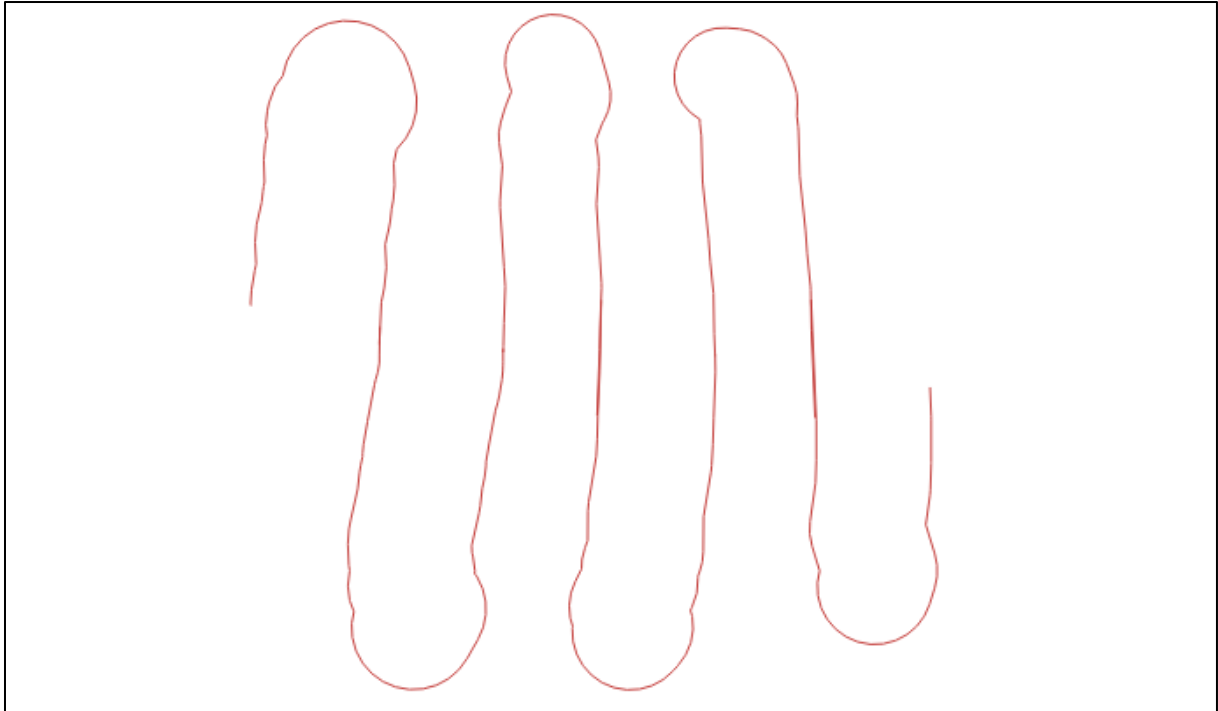
#### **4.7.6 Assemblage des virages**

Ensuite, nous assemblons les virages obtenus pour former une seule série (figure 4.15). On remarque que les virages résultants se chevauchent parfois et d'autres fois ils sont déconnectés (figure 4.14).



**Figure 4.14 Méthode d'assemblage de deux virages successifs. A gauche, on a des virages disjoints et à droite des virages sécants.**

Nous procédons alors de la manière suivante : Le premier virage n'est pas modifié. Considérons ensuite, deux virages consécutifs  $v_1$  et  $v_2$  dans cet ordre. Le dernier point de  $v_1$  (nommé  $p_1$ ) et le premier point de  $v_2$  (nommé  $p_2$ ) sont projetés sur le squelette de  $v_1$  appelé  $s_1$ . On obtient deux points de projection  $p_1'$  et  $p_2'$ . Pour les virages intersectants, on peut savoir que les virages se croisent si  $p_2'$  vient après  $p_1'$ . La fin de  $v_1$  est laissée telle quelle et le virage  $v_2$  est découpé par le segment  $p_1p_1'$ , puis le virage est translaté en  $p_1$ . Pour les disjonctions. On peut savoir que  $v_1$  et  $v_2$  sont disjoints si  $p_2'$  vient avant  $p_1'$ . Le point  $p_2'$  est projeté perpendiculairement du squelette  $s_1$  avec le paramètre du buffer du virage, on obtient le point  $p_2''$ . Ensuite,  $v_2$  est translaté en  $p_2''$ .



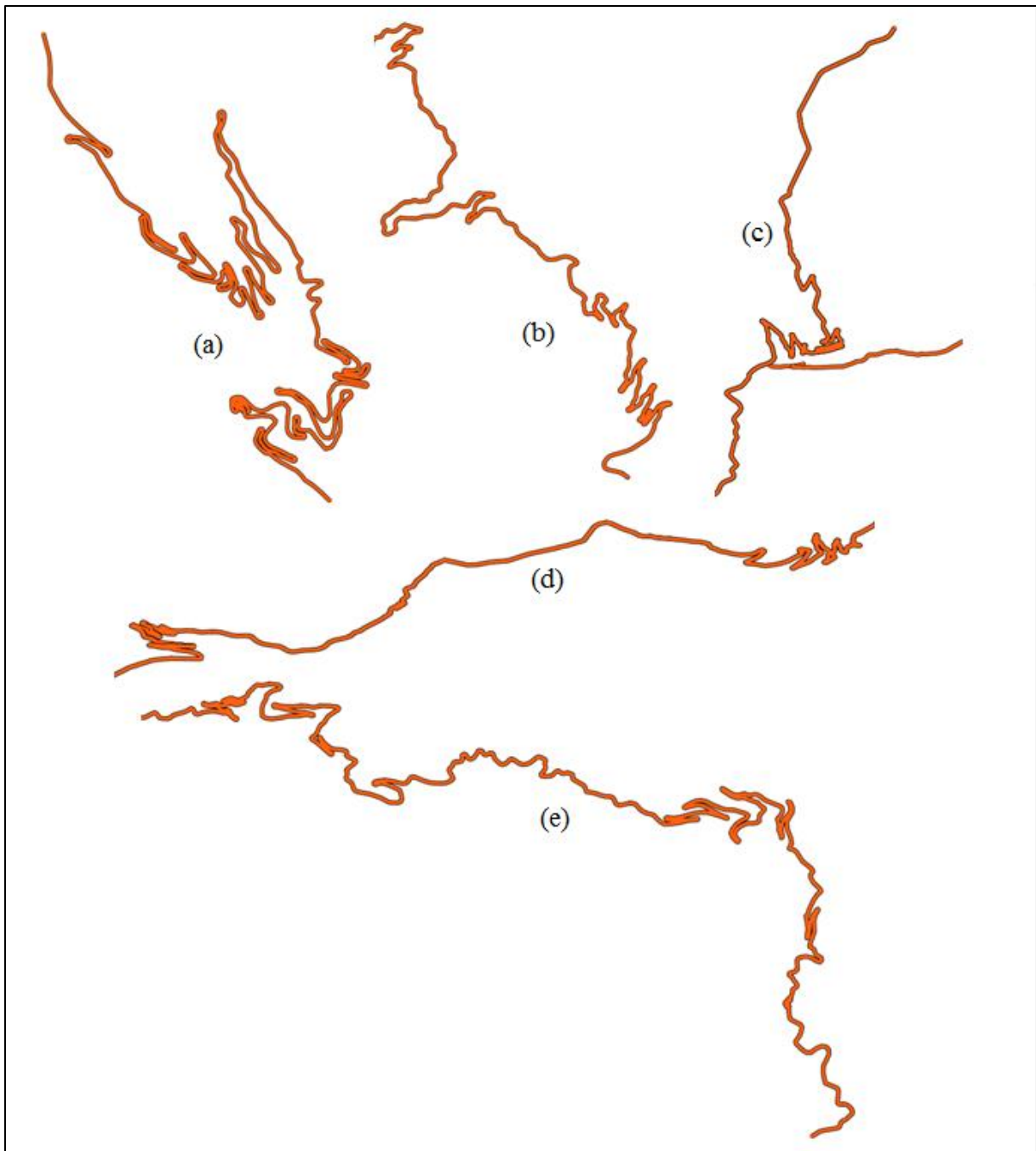
**Figure 4.15 La série de virage résultante**

#### **4.7.7 Réintégration de la série dans la route**

La série obtenue est translatée vers le milieu de l'axe de la ligne d'origine. Après cela, nous appliquons une rotation sur l'axe qui équivaut à la différence d'axe entre la ligne d'origine et la ligne résultante. Si la longueur de l'axe des séries a changé, la propagation des résultats est nécessaire. On peut déplacer la ligne de chaque côté de la série par translation ou on peut la tasser.

#### **4.8 Résultats**

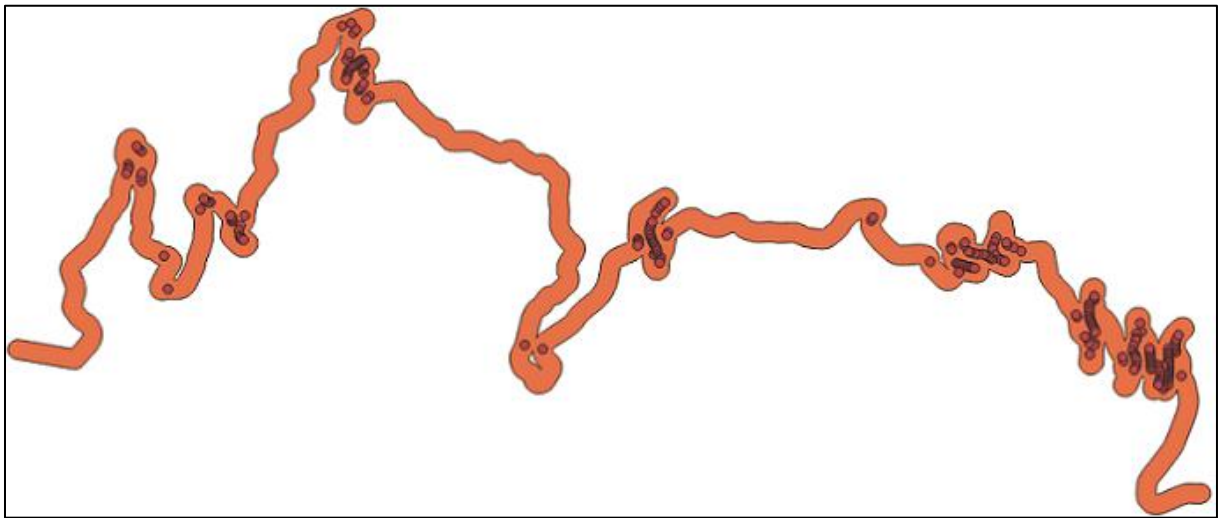
L'algorithme a été implémenté en langage PyQGIS (Python) sous le logiciel de cartographie QGIS. Les données proviennent d'OpenStreetMap. L'algorithme a été appliqué à des routes connues pour leur sinuosité dont les séries de virages ont été extraites (figure 4.16). Ces routes sont le Col de Chrea (Algérie), le Mont Tianmen (Chine), le col de Trollstigen (Norvège), le Col de la Furka (Suisse) et le Col de Turini (France).



**Figure 4.16** Echantillons de routes : Mont Tianmen, Chin (a), Col de Chrea, Algerie (b), Trollstigen, Norvège (c), Col de Furka in suisse (d) et Col de Turini, France (e).

Pour contrôler l'efficacité et l'efficacité de l'élimination de l'empatement, nous avons utilisé une mesure que nous avons développée pour vérifier qu'il n'y a plus d'empatement sur la ligne après application d'un algorithme de désempatement (figure 4.17). Le principe de cette mesure consiste à calculer un buffer autour de la ligne qui est égal à son symbole.

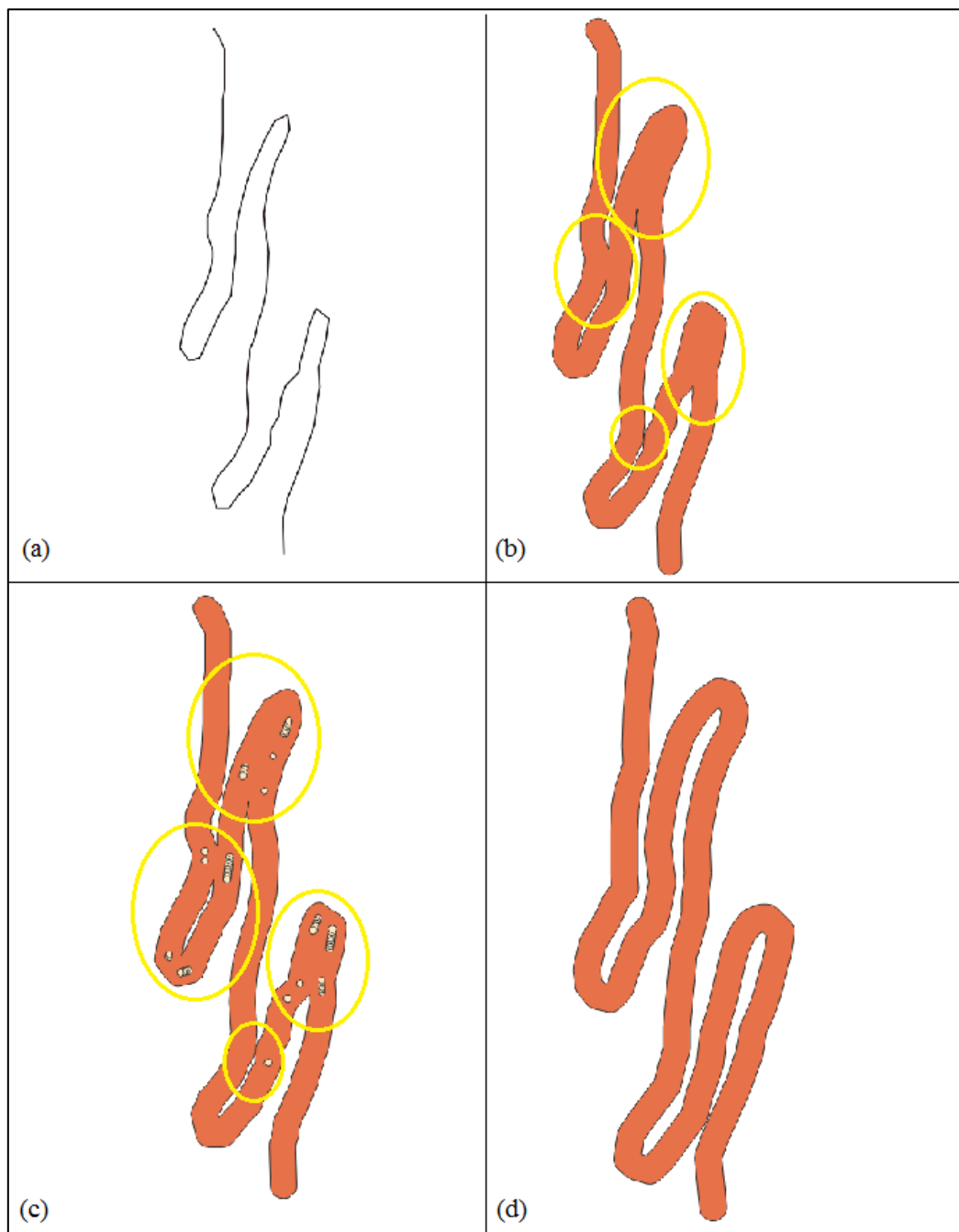
Ensuite, ce buffer est balayé par un cercle de rayon légèrement plus grand que le symbole. Si ce cercle n'est pas contenu dans le buffer, alors il n'y a pas d'empâtement à cet endroit, si le buffer contient le cercle, il y a alors un risque que la route soit empâtée en ce point. Le but est de n'avoir aucun point qui puisse représenter un empâtement. Le paramètre de l'algorithme (le rayon du cercle de balayage) peut changer en fonction de la façon dont nous définissons l'empâtement, la valeur que nous avons utilisée ci-dessous représente le minimum possible.



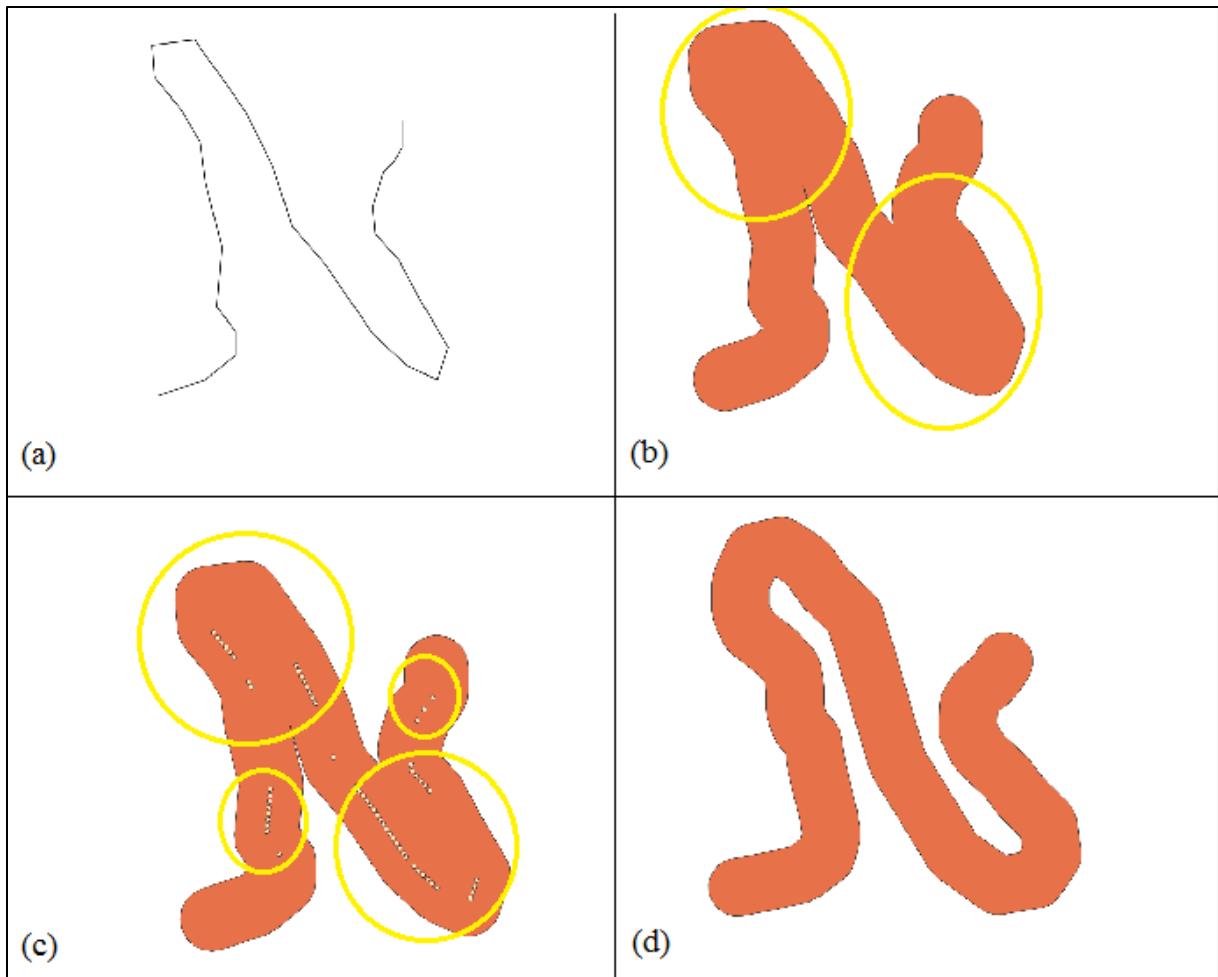
**Figure 4.17 La mesure d'empâtement.**

Voici quelques exemples d'échantillons de virages que nous avons testés (Figure 4.18, Figure 4.19 et Figure 4.20). Les virages sont étroits, serrés et sinueux et l'axe de la série n'est pas droit. L'image contient d'abord la ligne sans symbolisation (Figure 4.18(a), Figure 4.19(a) et Figure 4.20(a)). Ensuite, nous avons la ligne symbolisée (Figure 4.18(b), Figure 4.19(b) et Figure 4.20(b)), nous pouvons remarquer l'apparition de l'empâtement (entouré par des cercles jaunes). On a alors le résultat de l'application de la mesure d'empâtement (Figure 4.18(c), Figure 4.19(c) et Figure 4.20(c)), on peut voir l'apparition de points d'empâtement indiquant la présence de l'empâtement (entourés par des cercles jaunes). Enfin, nous avons le résultat de l'application de l'algorithme MBRS sur lequel nous avons appliqué la mesure d'empâtement (Figure 4.18(d), Figure 4.19(d) et Figure 4.20(d)), nous pouvons observer la

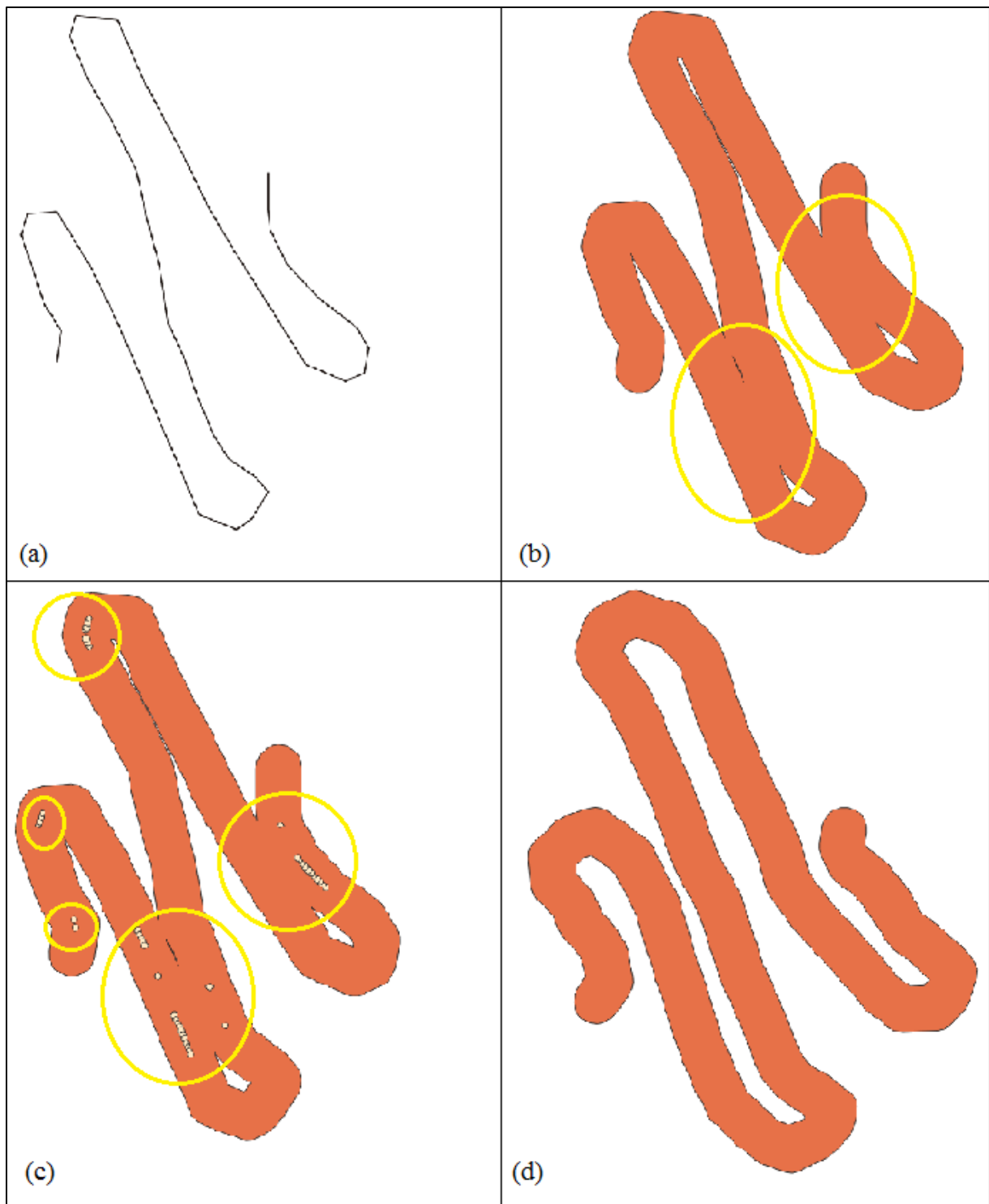
disparition des points d'empatement, nous pouvons donc conclure que notre algorithme a pu éliminer l'empatement de la série.



**Figure 4.18 Résultat n °1, (a) est la série originale, (b) est la série après symbolisation, (c) est l'application de la mesure d'empatement sur la série symbolisée, (d) est l'application de la mesure d'empatement après traitement de la série par MBRS.**

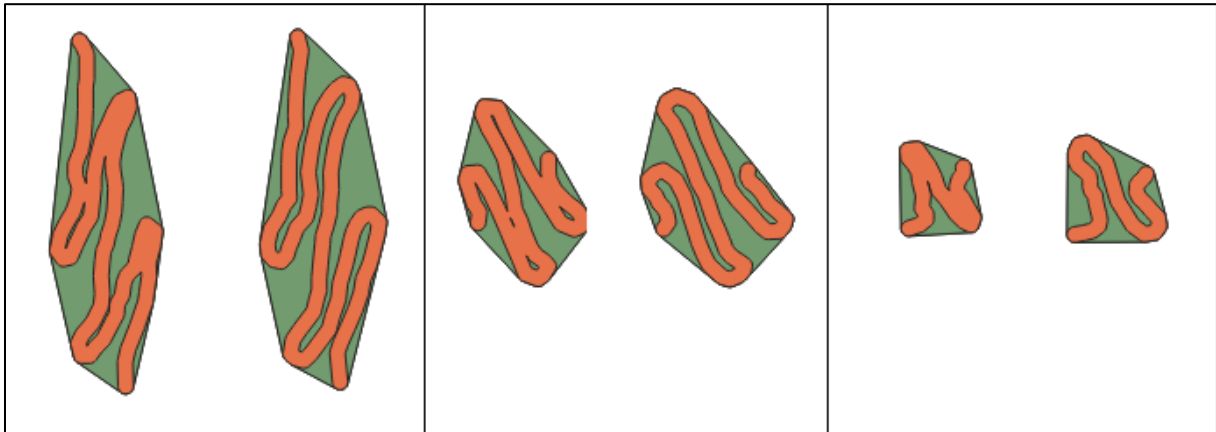


**Figure 4.19 Résultat n ° 2, (a) est la série originale, (b) est la série après symbolisation, (c) est l'application de la mesure d'empâtement sur la série symbolisée, (d) est l'application de la mesure d'empâtement après traitement de la série par MBRS.**



**Figure 4.20 Résultat n ° 3, (a) est la série originale, (b) est la série après symbolisation, (c) est l'application de la mesure d'empâtement sur la série symbolisée, (d) est l'application de la mesure d'empâtement après traitement de la série par MBRS.**

Des mesures ont été réalisées pour savoir si le résultat produit variait l'étendue terrain par rapport à la valeur d'origine (figure 4.21). Nous avons utilisé l'enveloppe convexe et la longueur de l'axe. L'enveloppe convexe est la surface convexe minimale recouvrant l'objet, la longueur de l'axe est constituée de la distance entre les deux extrémités de la ligne.



**Figure 4.21** La mesure de l'enveloppe convexe.

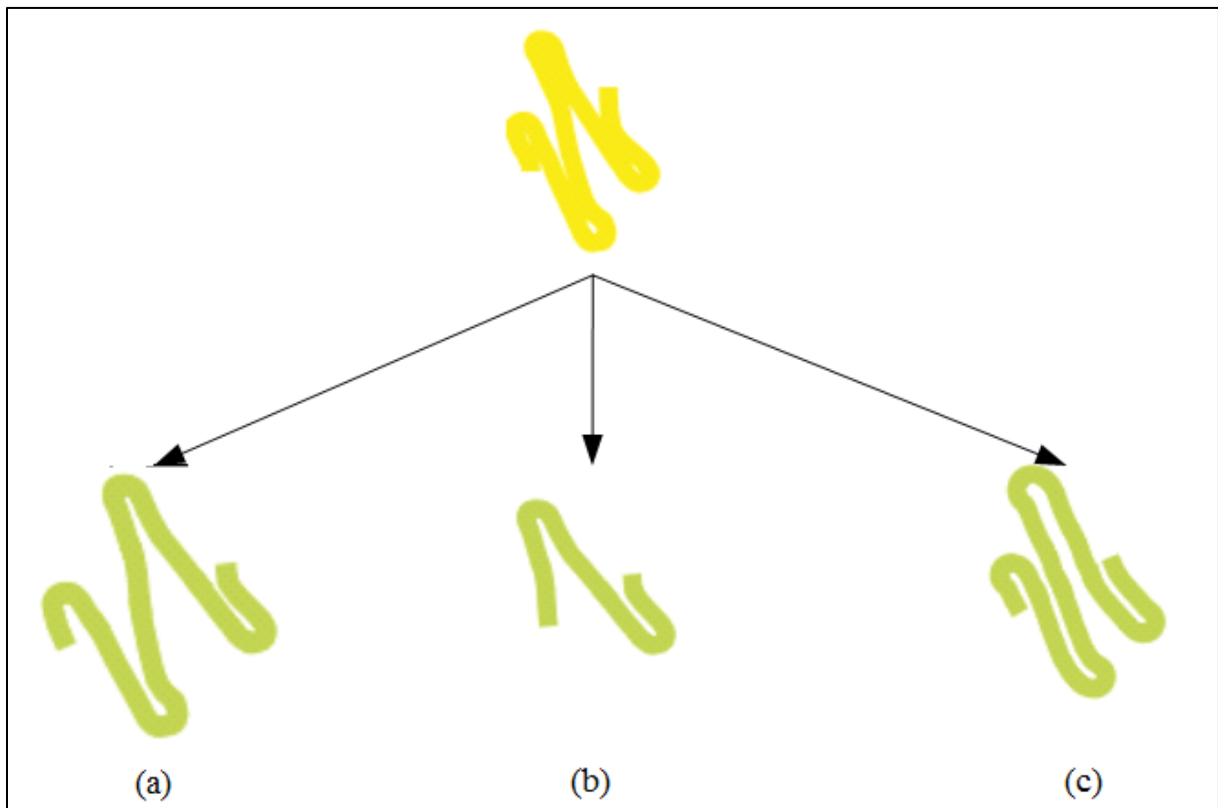
Les valeurs données sont un rapport de la valeur résultante et la valeur d'origine et sont données en pourcentage. On note une augmentation moyenne de 16% dans l'axe de la ligne et une augmentation moyenne de 25% dans l'enveloppe convexe.

Au niveau du virage, l'algorithme supprime l'empâtement. Il est efficace sur les longs virages étroits. Il permet de conserver la forme générale tout en limitant l'emprise terrain. Au niveau de la série, l'algorithme ne souffre pas de la déformation de l'axe routier. Il permet également de garder l'harmonie des tailles des différents virages.

#### **4.9 Comparaison**

Les algorithmes existants les plus efficaces sur les séries de virages routiers sont Accordéon [68] et Schématisation [51]. Nous avons donc choisi ces algorithmes pour notre comparaison. Nous pouvons voir dans la figure 4.22 suivante le résultat de l'application des trois algorithmes sur une série de virages routiers, Figure 4.22(a) représente l'application de

l'algorithme Accordéon, Figure 4.22(b) est l'application de l'algorithme Schématisation et Figure 4.22(c) est l'application de l'algorithme MBRS.



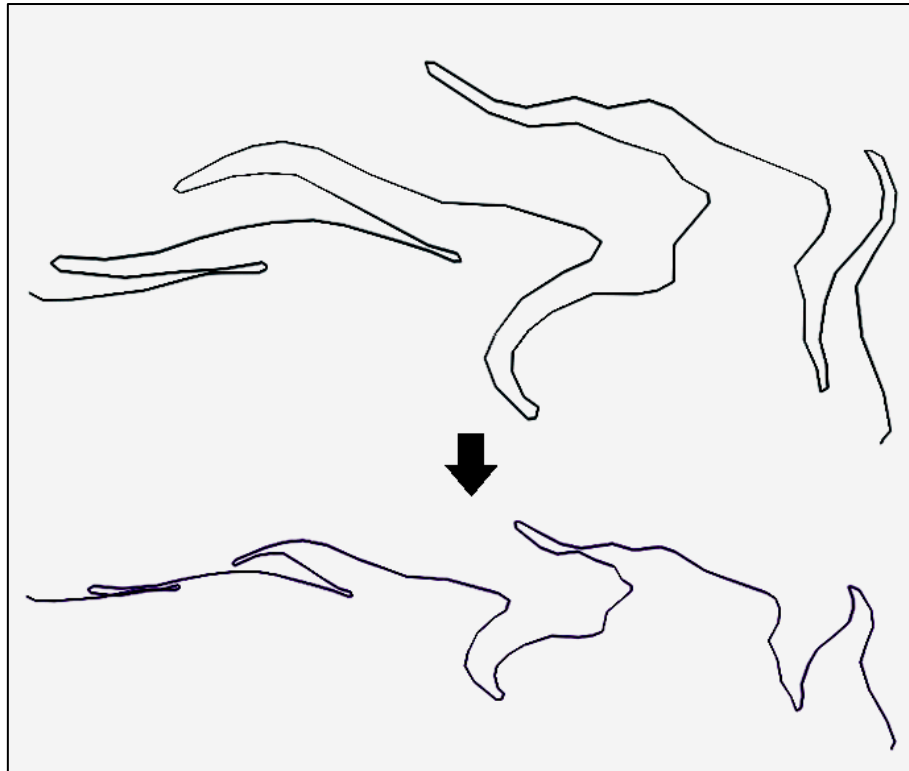
**Figure 4.22 Comparaison, (a) est le traitement de la série par Accordéon, (b) est le traitement de la série par Schématisation, (c) est le traitement de la série par MBRS.**

Nous avons sélectionné quatre mesures pour la comparaison: la longueur de l'axe qui se compose de la distance entre les deux extrémités de la ligne, la surface de l'enveloppe convexe, la distance de Hausdorff [1] qui est la distance maximale entre la ligne d'origine et la ligne résultante ainsi que le nombre de virages (Tableau 4.1).

**Tableau4. 1: Comparaison de MBRS avec Accordéon et Schématisation**

L'algorithme	Accordéon	Schématisation	MBRS
La longueur d'axe	+70%	+0%	+16%
L'enveloppe convexe	+56%	-37%	+25%
La distance de Hausdorff	25	44	20
Le nombre de virages	4	2	4

Les valeurs sont données par rapports aux originales et elles sont données au moyen de pourcentages. Le tableau ci-dessus représente l'évaluation des résultats des trois algorithmes. Il est à noter que le L'algorithme d'accordéon conserve le nombre initial de virages, cependant, il provoque une augmentation de 70% de la longueur de l'axe initial ainsi qu'une augmentation de l'enveloppe qui peut causer des problèmes sur la carte avec les objets environnants. Parfois, Accordéon peut entraîner un aplatissement des virages ou des erreurs topographiques si l'axe de la série n'est pas rectiligne ou que les virages sont orientés parallèlement à l'axe d'élargissement (figure 4.23).



**Figure 4.23 Problématique d'Accordéon.**

Pour la schématisation, aucun changement n'est perceptible par rapport à la longueur de l'axe résultant, par opposition à l'enveloppe qui diminue de 37% en taille, mais il réduit le nombre de virages et il a la plus grande valeur de la distance de Hausdorff, car il apporte les changements les plus significatifs à la forme de la ligne. L'algorithme MBRS fait en sorte que le nombre de virages reste le même et augmente modérément la longueur de l'axe et de l'enveloppe.

#### **4.10 Synthèse et Conclusion**

Les trois algorithmes ont pu corriger l'empâtement routier sur les séries. Ils restent complémentaires, et le choix de l'algorithme peut se porter selon les types de virages traités et l'étendue terrain disponible ainsi que la densité de la zone traitée (s'il y a beaucoup d'objets aux alentours ou si c'est clairsemé). Accordéon et Schématisation sont efficaces sur les séries de virages simples mais le premier est à favoriser si l'espace disponible est grand et le second est à favoriser sinon. MBRS est efficace si la série de virages traitée est complexe et la densité de la zone grande.

## Chapitre 5 : Proposition d'une mesure de détection de patterns de bâtiments répétés

Dans ce chapitre, nous présentons notre contribution dans le domaine du bâti. Nous y détaillons donc une nouvelle mesure de détection des patterns de bâtiments.

## 5.1 Introduction

Dans ce chapitre nous allons tout d'abord décrire une nouvelle méthode de détection de patterns de bâtiments [41], cette méthode peut être classée comme mesure de généralisation. Son objectif est de détecter des patterns de bâtiments qui se répètent comme le motif dans un tissu (figure 5.1). Avant de voir la méthode, voyons ce qu'est un pattern.



Figure 5.1 Exemples de Patterns de bâtiments.

## 5.2 Définition d'un pattern

Nous pouvons définir un pattern de plusieurs façons, [99] définit le pattern comme un groupe de bâtiments voisins ayant les mêmes propriétés. "Nous définissons des patterns de bâtiments au niveau d'un cluster d'objets spatialement proches avec une géométrie similaire (par exemple: espacement, taille, orientation et forme) et des propriétés sémantiques ... Au niveau suivant, nous nous référons à la construction de clusters dans des alignements linéaires et non linéaires. Grilles en termes de forme de groupe. " [97] voit les patterns comme des bâtiments disposés linéairement ou disposés sous la forme de lettres de l'alphabet Z, H ou en escalier. On peut déduire des deux définitions précédentes, la définition suivante: un motif est un ensemble de bâtiments de propriétés similaires impliqués dans un agencement particulier: alignement, grille, étoile, cercle, carré, Z, H, etc. Un contre-exemple serait des bâtiments dispersés dont les propriétés sont différentes.

Dans la littérature, il existe d'autres définitions du mot pattern (motif) "Un motif est un objet original utilisé pour faire des copies, ou un ensemble d'objets répétitifs dans un dessin décoratif" [94] ou "C'est souvent un phénomène ou une organisation qui peut être observé à plusieurs reprises dans l'étude de certains sujets, auxquels nous pouvons conférer des propriétés caractéristiques. "

On peut se poser la question de savoir pourquoi un alignement est un motif? C'est parce que chaque groupe qui reproduit les mêmes caractéristiques qu'un alignement est un alignement. Il en va de même pour une grille, un cercle, etc. Mais pour un motif spécifique à une carte qui ne suit pas une forme connue, que pourrions-nous dire? Par exemple, un groupe de deux, trois ou quatre bâtiments pentagonaux qui se font face sur une carte, de manière répétée. Nous pourrions choisir de dire dans ce cas qu'un motif est un ensemble de groupes de bâtiments ayant les mêmes caractéristiques de forme et de disposition. Nous partons de cette définition pour définir notre algorithme. Dans ce qui suit, nous choisissons de nommer modèle le dessin du motif, l'occurrence du motif les groupes reproduisant le modèle du motif et le motif l'ensemble des occurrences d'un motif.

### **5.3 La méthode**

Avant de présenter la méthode, il est important de spécifier certaines propriétés et règles pour nos modèles. Ces dernières peuvent être modifiées et adaptées par la suite:

1. Un bâtiment ne peut pas être inclus dans plus d'un motif ou occurrence d'un motif. Deux motifs ou occurrences de motifs sont à 100% distincts.
2. Chaque bâtiment d'une occurrence a des bâtiments correspondants dans les autres occurrences de ce même modèle. Ils sont équivalents en forme, taille, position, distance et orientation (relativement parlant).
3. Une occurrence de motif équivaut à une autre occurrence du même motif orienté de 0 à 360° ou à son miroir orienté de 0 à 360°.

En d'autres termes, les occurrences d'un motif doivent être empilables comme des feuilles. Cependant, ces règles ne concernent pas les homothéties de motifs. Pour les homothéties de modèle, les distances diffèrent d'un facteur  $k$  fixe.

Notre méthode consiste à créer des groupes de bâtiments voisins de différentes tailles et à rechercher des similitudes entre ces groupes. Les étapes sont décrites dans le diagramme ci-dessous (Figure 5.2):

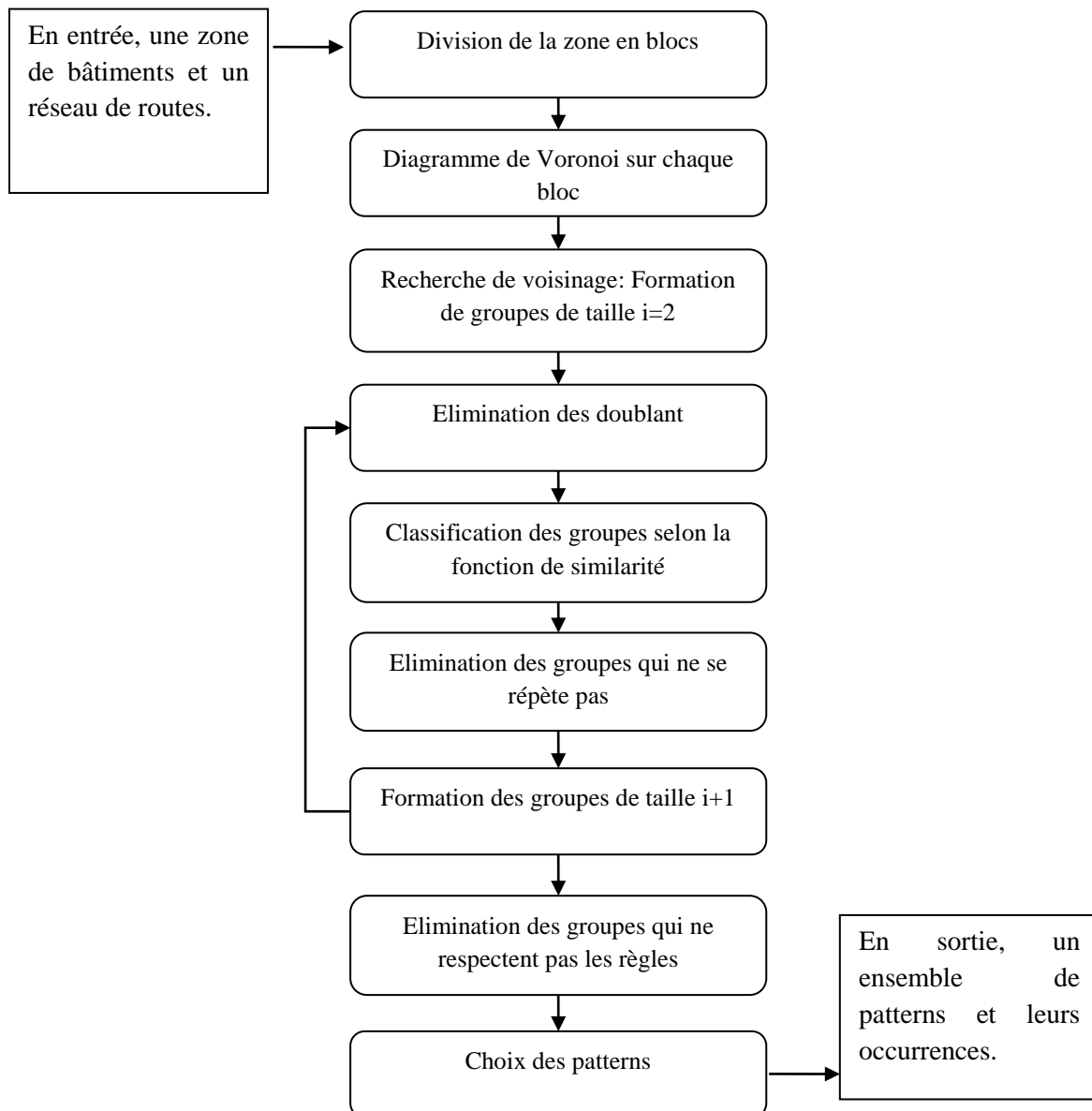
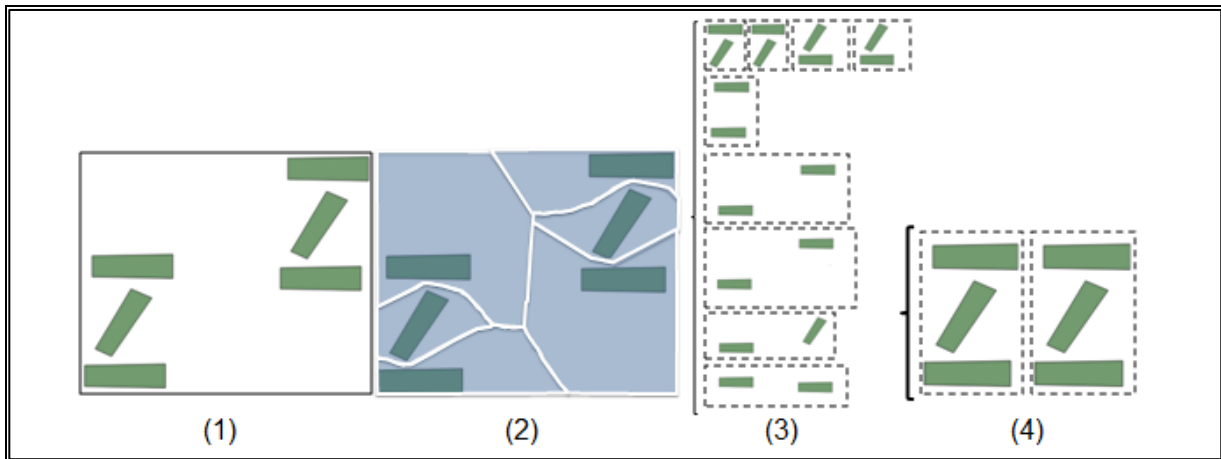


Figure 5.2 Le processus d'extraction des motifs.

1. Division de la zone en blocs: Elle consiste à diviser la zone géographique selon le réseau de routes et de rues. Ces sous-zones sont appelées blocs.
2. Recherche de voisinage: un diagramme de Voronoi est appliqué à chaque bloc. Cela nous a permis de regrouper les bâtiments en paires de bâtiments, où chaque couple  $(b_i, b_j)$  signifie que le bâtiment  $b_i$  est adjacent au bâtiment  $b_j$  et vice-versa. Nous appelons l'ensemble des groupes de taille deux,  $S_2$ .
3. Élimination de la duplication:  $(b_i, b_j)$  est similaire à  $(b_j, b_i)$ .
4. Classification des groupes: Nous recherchons des groupes similaires en utilisant la fonction de similitude. Nous regroupons ces groupes. Ceux-ci représentent des patterns potentiels. Nous appelons l'ensemble des patterns potentiels de tailles 2,  $PP_2$ .
5. Élimination des groupes non répétés (dont le nombre d'occurrences est égal à 1) des ensembles  $S_2$  et  $PP_2$ .
6. Formation de groupes de plus grande taille, c'est-à-dire, de taille 3 à partir de l'ensemble  $S_2$ . Nous recherchons deux groupes de taille deux ayant un bâtiment en commun et nous les fusionnons. On obtient alors l'ensemble  $S_3$ .
7. Les étapes 3 à 6 sont ensuite répétées avec  $S_i$  et  $PP_i$  l'ensemble des groupes et des motifs potentiels de tailles  $i$ . Pour l'étape 6, si nous avons des groupes de taille  $i$  et que nous voulons former des groupes de taille  $i + 1$ , nous regardons par groupes de taille  $i$  ceux ayant  $i-1$  bâtiments communs.
8. On s'arrête lorsque l'ensemble  $PP_i$  obtenu est égal à l'ensemble vide, c'est-à-dire qu'il n'y a plus de groupes qui se répètent.
9. On obtient les ensembles  $P_i$  qui représentent les motifs de tailles  $i$ . Ceci en éliminant les groupes de  $PP_i$  qui ne respectent pas les règles sur les motifs mentionnés plus haut.
10. Nous choisissons les motifs qui ont le plus grand nombre de bâtiments et le plus petit nombre d'occurrences. Quand il y a une multitude de motifs sur les mêmes bâtiments.

L'image ci-dessous (Figure 5.3) illustre les étapes de la méthode :



**Figure 5.3 La méthode de groupage. (1) représente les données d'origine. (2) application d'un diagramme de Voronoi. (3) voisinage, groupes de taille 2 à l'itération 1. (4) groupes de taille 3 à l'itération 2.**

Le but de la fonction de similitude est de comparer deux groupes de bâtiments (G1 - G2) pour vérifier s'il s'agit d'occurrences du même modèle. Il repose sur trois opérations géométriques: la translation, la rotation et la mise en miroir pour voir si les deux groupes sont empilables. Nous translatons d'abord le premier groupe de son point central au centre du deuxième groupe, puis nous commençons la rotation jusqu'à ce que les deux groupes soient empilés, sinon nous procédons à une mise en miroir du premier groupe et ensuite nous le tournons jusqu'à ce que les deux groupes soient empilés. Pour la rotation, nous prenons simplement les angles formés par le point central de chaque bâtiment du deuxième groupe et le point central de ce même groupe, ainsi nous ne prenons en compte que quelques positions qui correspondent au nombre de bâtiments du groupe G2. La méthode est illustrée dans la figure 5.4, les formules mathématiques sont décrites ci-dessous :

Translation: La translation de G1 en G2 est donnée par ensemble d'équations 5.1.

$$x' = x + x'_0 - x_0 \quad (5.1)$$

$$y' = y + y'_0 - y_0$$

$c1 (x_0, y_0)$ ,  $c2 (x_0', y_0')$  sont les centres de gravité de G1 et G2 respectivement.

Miroirisation: La mise en miroir de G1 par la droite  $x = x_0$  (passant par  $c1$ ) est donnée par ensemble d'équations 5.2.

$$x' = 2 \times x_0 - x \tag{5.2}$$

$$y' = y$$

Rotation: On utilise les coordonnées polaires concernant  $c1$ , une rotation de G1 de l'angle  $\alpha$  est donnée par ensemble d'équations 5.3.

$$x' = x_0 + r \times \cos(\beta + \alpha) \tag{5.3}$$

$$y' = y_0 + r \times \sin(\beta + \alpha)$$

$r$ : la distance entre un point et  $c1$

$\beta$  est donné par la fonction arc tangente en  $2\pi$ .

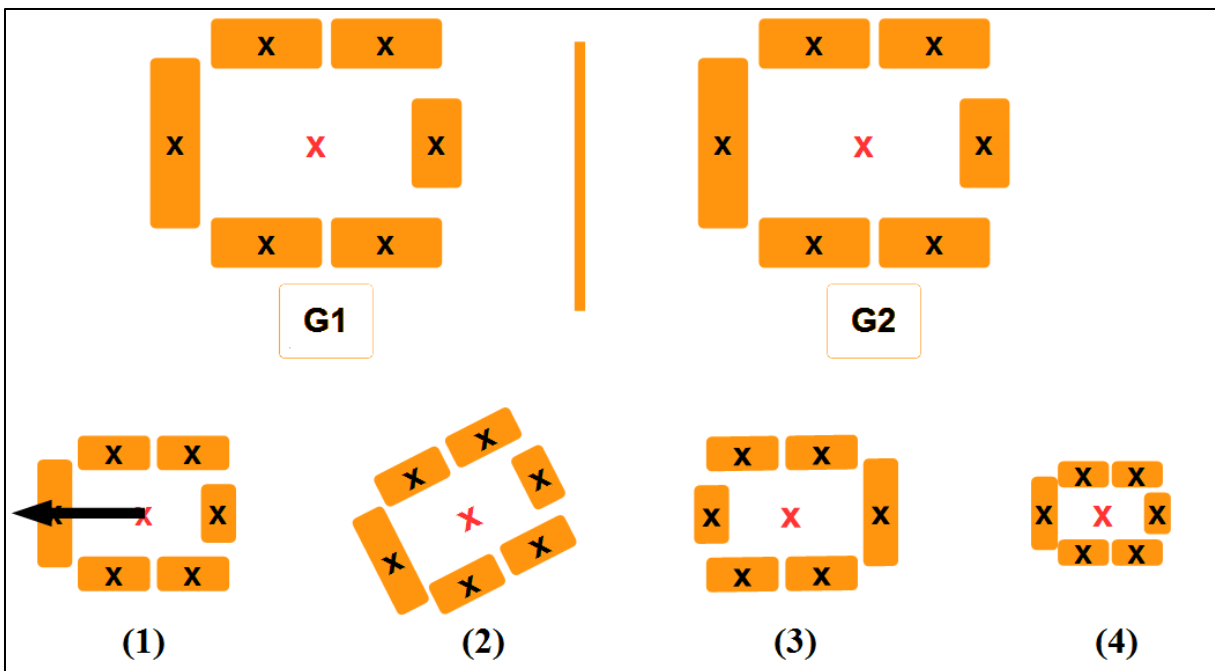


Figure 5.4 La fonction de similarité

## 5.4 Résultats

Pour la mise en œuvre du projet, nous avons choisi le logiciel QGIS. Il dispose des outils SIG habituels. De plus, il propose le langage de programmation PyQGIS qui est un langage python enrichi des bibliothèques spatiales QGIS. Les données proviennent d'OpenStreetMap et de GoogleMaps. Les échantillons de données proviennent de la carte de la ville de Dubaï. Dans la figure 5.5(a), nous voyons la zone géographique sur laquelle les tests ont été effectués, nous avons donc une zone urbaine couverte de bâtiments et de rues. Dans la figure 5.5(b), nous avons la division de la zone en blocs. La figure 5.5(c) montre l'application du diagramme de Voronoï pour objets sur un bloc de la zone et la figure 5.5(d) montre les motifs détectés dans le bloc. On note (visuellement) que l'algorithme a bien détecté les différents patterns et leurs occurrences, ces derniers étant bien isolés les uns des autres.

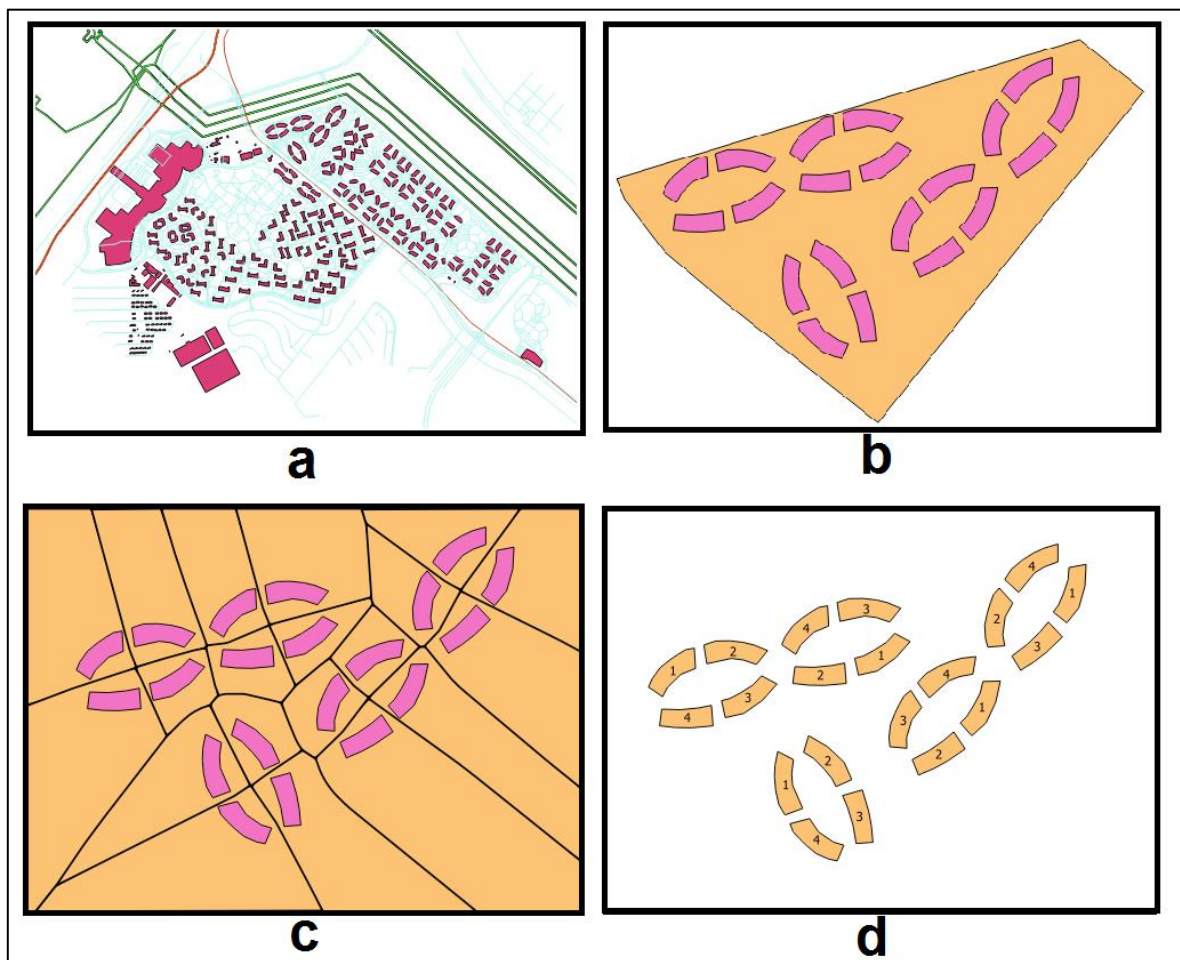
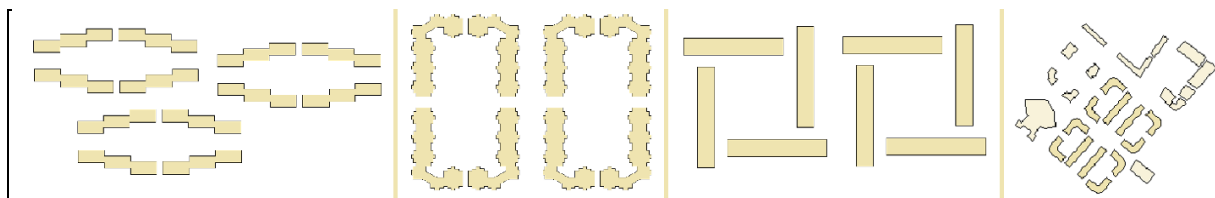


Figure 5.5 Résultat

Le test de validation a été réalisé sur la région saharienne d'Algérie (Laghouat) et a confirmé les résultats précédemment obtenus sur l'échantillon de Dubaï. On peut voir ci-dessous (Figure 5.6) les différents échantillons de pattern.



**Figure 5.6 Validation**

La précision des résultats dépend de la précision lors de la saisie des données. La détection est en fonction d'une marge d'erreur en paramètre, si elle est trop grande, on aura une sur-détection et si elle est trop petite, la détection sera limitée. C'est un facteur d'échelle. Il y a aussi parfois des motifs qui peuvent se ressembler à l'œil nu mais leurs tailles ne sont pas exactement les mêmes en réalité.

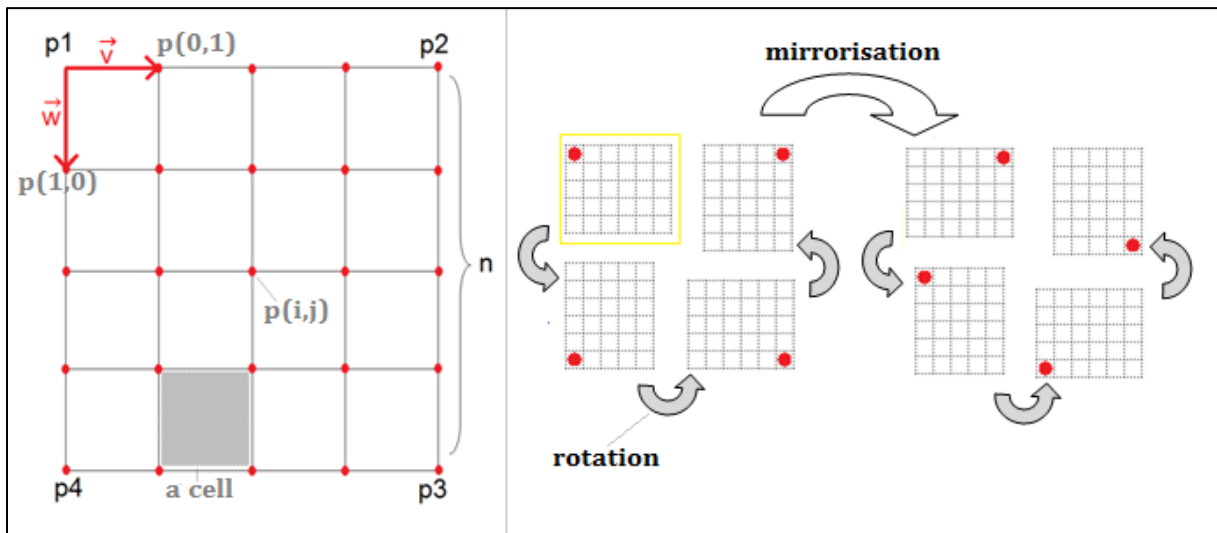
## 5.5 Amélioration de la méthode

Cette méthode a été améliorée pour détecter plus de motifs et permettre l'ajout de nouveaux motifs à détecter d'une manière plus simplifiée. Ceci est en réponse aux perspectives de la méthode précédente.

La méthode de groupage est restée la même. La fonction de similarité est la suivante: pour chaque groupe, nous calculons le rectangle englobant orienté. Ce rectangle est ensuite divisé en cellules comme une grille. La taille de la grille est un paramètre de la fonction. Nous vérifions ensuite pour chaque cellule si elle croise les bâtiments du groupe. Si c'est le cas, la cellule d'un tableau de même taille que la grille correspondante est équivalente à 1 sinon elle équivaut à 0. On obtient ainsi une matrice de 0 et de 1 pour chaque groupe. Il sera plus facile de comparer deux groupes en comparant leurs matrices. Il y a huit positions à comparer (Figure 5.7), les quatre rotations possibles de la matrice, le miroir et les trois rotations de ce

dernier. L'homothétie est induite implicitement puisque chaque rectangle s'adapte à son groupe et on garde le même nombre de cellules. La comparaison se fait par test d'égalité cellule à cellule. L'extraction du maillage se fait comme suit: soit  $g$  un groupe de bâtiments, soit  $\{p_1, p_2, p_3, p_4\}$  les quatre points du rectangle englobant orienté de  $g$  nommé  $r$ . Nous voulons diviser  $r$  en  $n \times n$  cellules. On prend  $p_1$  comme point de référence, les coordonnées du point  $p$  ( $i, j$ ) sont données par l'ensemble d'équations 5.4.

$$\begin{aligned}
 p_{0,1} &= \frac{n-1}{n}p_1 + \frac{1}{n}p_2 \\
 p_{1,0} &= \frac{n-1}{n}p_1 + \frac{1}{n}p_4 \\
 \vec{v} &= p_{0,1} - p_1 \\
 \vec{w} &= p_{1,0} - p_1 \\
 p_{i,j} &= p_1 + i\vec{v} + j\vec{w} \\
 cell_{i,j} &= \{p_{i-1,j-1}, p_{i-1,j}, p_{i,j}, p_{i,j-1}\} \\
 p_{i,j} &= p_{i,n-j+1} \\
 p_{i,j} &= p_{j,n-i+1}
 \end{aligned} \tag{5.4}$$



**Figure 5.7** Calcul de la matrice d'un groupe de bâtiments sur l'image de gauche et traitement matriciel sur l'image de droite (Rotation et Miroir) il y a huit matrices possibles.

Si l'intention n'était pas de rechercher un motif répété mais un modèle particulier de type géométrique (losange, cercle) ou lettre de l'alphabet (Z, H), il serait facile de garder son modèle (sa matrice) en mémoire et de le rechercher dans la couche de bâtiments. Ainsi, il est possible d'enrichir facilement la bibliothèque de modèles pour détecter plus de modèles. Cependant une étape de Machine Learning est nécessaire. La figure Figure 5.8 illustre les matrices de différents modèles de patterns.

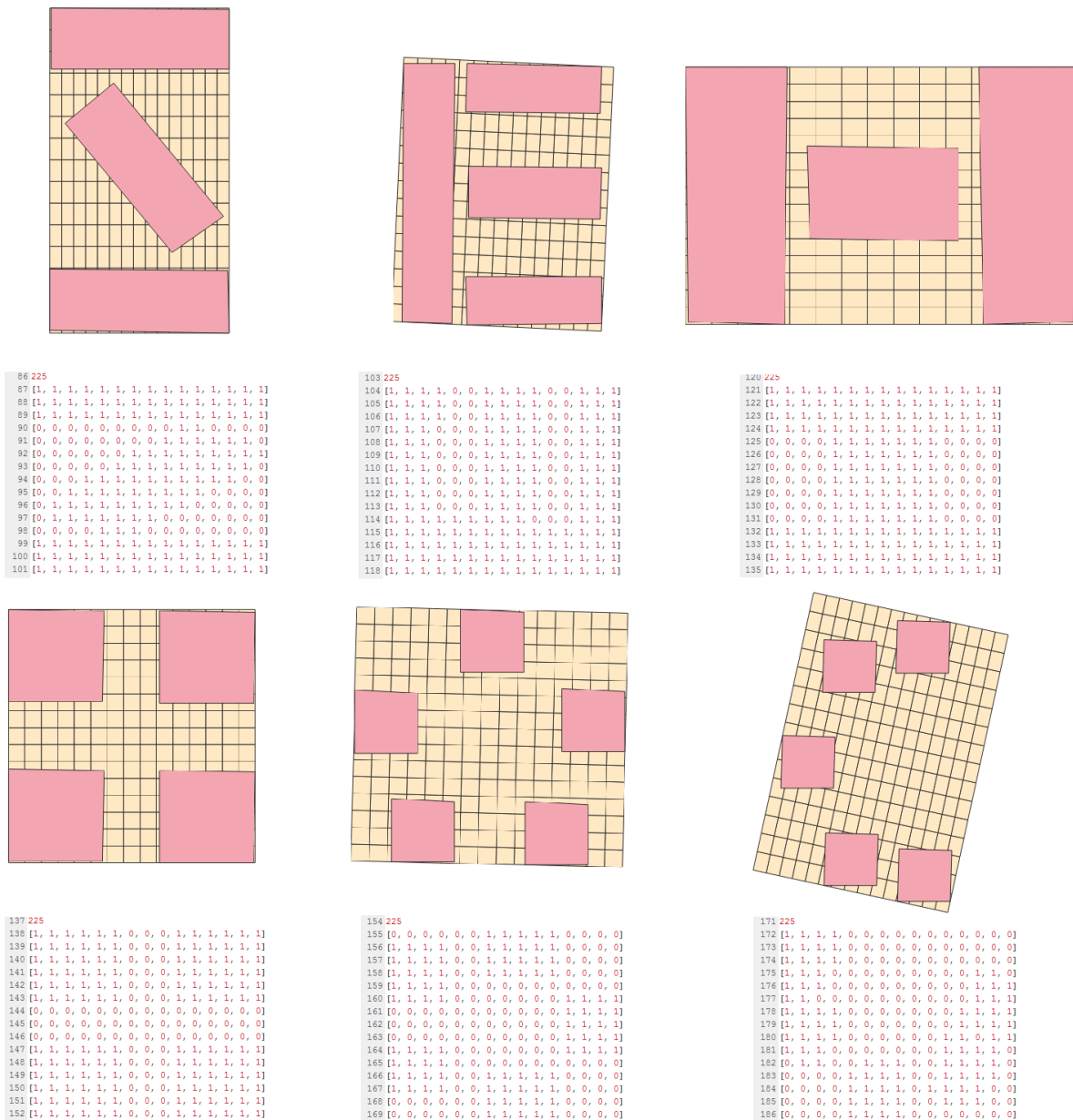
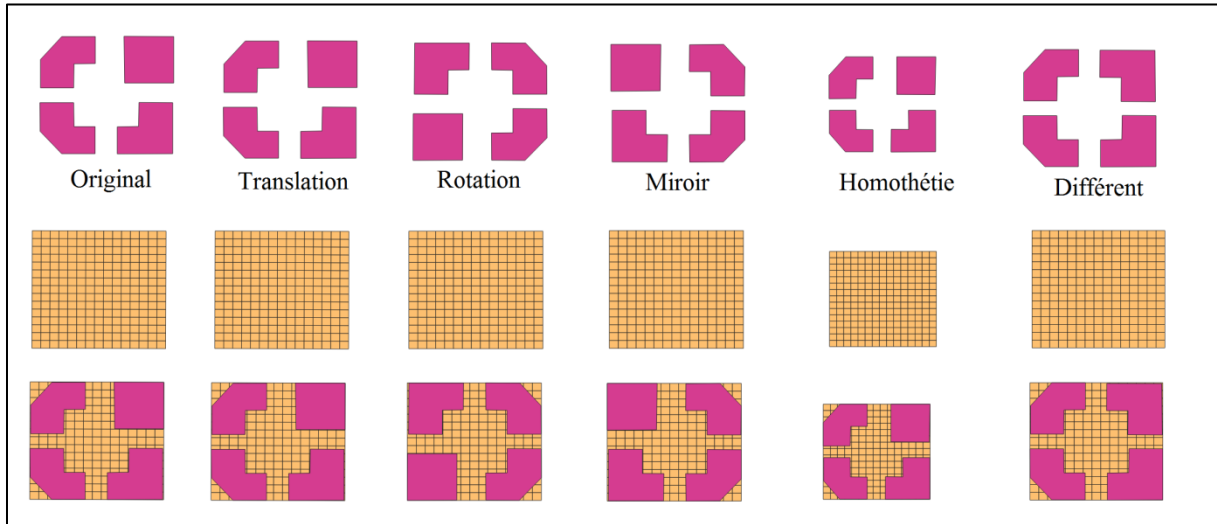


Figure 5.8 Matrices de modèles particuliers.

### 5.5.1 Analyse

Des tests sur des échantillons dans les différentes directions ont été effectués. Un modèle différent a été rajouté pour servir de témoin (Figure 5.9).

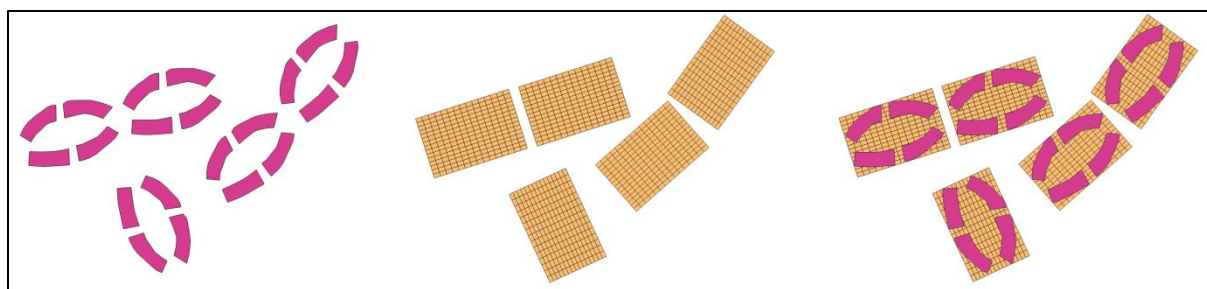


**Figure 5.9 Des groupes de bâtiments et leurs rectangles englobant orientés décomposés en cellules.**

Le tableau 5.1 ci-dessous illustre différents tests réalisés avec différents paramétrages de la taille de la matrice de comparaison. On remarque que plus la matrice est affinée plus la méthode est précise. On remarque aussi que si la taille de la matrice est petite on se retrouve avec une sur-identification.

**Tableau 5.1 Tests réalisés avec différents paramétrages de la taille de la matrice.**

Taille	Translation	Rotation	Miroir	Homothétie	Différent
4x4	Vrai	Vrai	Vrai	Vrai	Vrai
6x6	Vrai	Vrai	Vrai	Vrai	Faux
10x10	Vrai	Vrai	Vrai	Vrai	Faux
15x15	Vrai	Vrai	Vrai	Vrai	Faux



**Figure 5.10** Des patterns de bâtiments et leurs rectangles englobant orientés décomposés en cellules

La figure 5.10 représente un groupe d'occurrences d'un même pattern. Une occurrence est prise comme modèle. Les autres lui sont comparés dans toutes ses orientations avec différentes tailles de matrice. Le tableau 5.2 démontre cette comparaison.

**Tableau 5.2** Nombre de cellules différentes selon la taille de la matrice de comparaison.

	Tel	Rot1	Rot2	Rot3	Mir	Rot1	Rot2	Rot3
7x7								
G1	0	6	6	6	6	2	4	6
G2	8	6	4	4	8	6	6	2
G3	3	5	7	5	7	1	7	5
G4	5	7	5	3	5	7	1	7
8x8								
G1	0	12	4	12	4	10	4	12
G2	12	4	12	0	12	4	10	4
G3	12	0	12	4	10	4	12	4
G4	0	12	4	12	4	10	4	12
9x9								
G1	0	14	6	14	6	14	2	14
G2	15	5	15	1	15	5	15	3
G3	15	5	15	5	15	5	15	3
G4	3	17	5	17	5	17	3	17
10x10								
G1	0	18	10	18	6	18	6	18
G2	16	8	16	2	16	6	16	6
G3	17	3	17	7	17	5	17	7
G4	2	16	8	16	6	16	6	16
11x11								
G1	0	20	10	20	6	18	8	22
G2	19	11	19	1	21	7	17	9

G3	21	7	23	13	19	11	23	9
G4	5	21	13	19	9	17	11	21
12x12								
G1	0	36	10	36	6	34	10	34
G2	35	9	37	3	35	7	35	9
G3	38	4	36	10	34	10	36	8
G4	4	36	10	38	8	34	10	36
13x13								
G1	0	38	14	38	10	38	10	38
G2	38	16	38	4	38	10	38	12
G3	38	10	38	14	38	12	38	8
G4	6	40	18	40	10	40	12	40
15x15								
G1	0	58	26	58	14	56	26	58
G2	55	25	55	7	55	17	53	23
G3	54	12	54	26	52	22	54	18
G4	12	54	26	54	18	52	22	54

Tableau 5.3 Taux d'erreur (%)

7x7	8x8	9x9	10x10	11x11	12x12	13x13	14x14
4%	0%	3.7	3	5.7	2.7	4.7	5.3

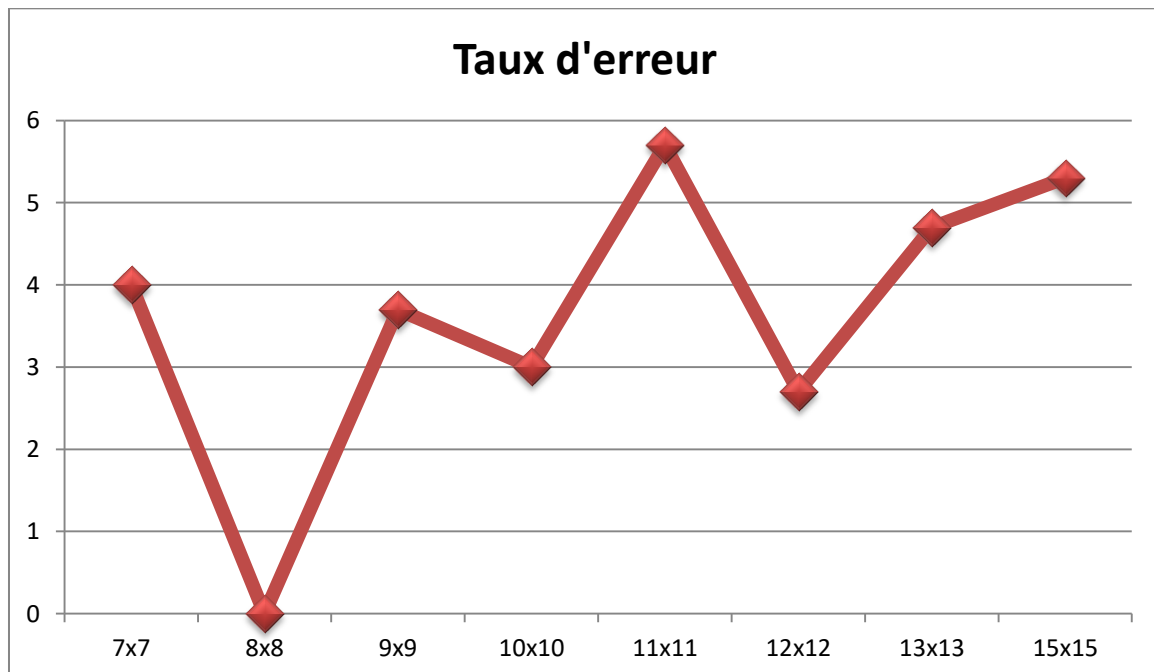


Figure 5.11 Graphique du taux d'erreur

On remarque selon le tableaux 5.3 et le graphique illustré dans la Figure 5.11 que la méthode a un meilleur paramètre (8x8). On remarque aussi une fluctuation du taux d'erreur, cependant légère. On peut en déduire que si la taille de la matrice est trop grande la méthode devient trop restrictive. Il est possible de palier à cette restriction en acceptant un certain taux d'erreur raisonnable.

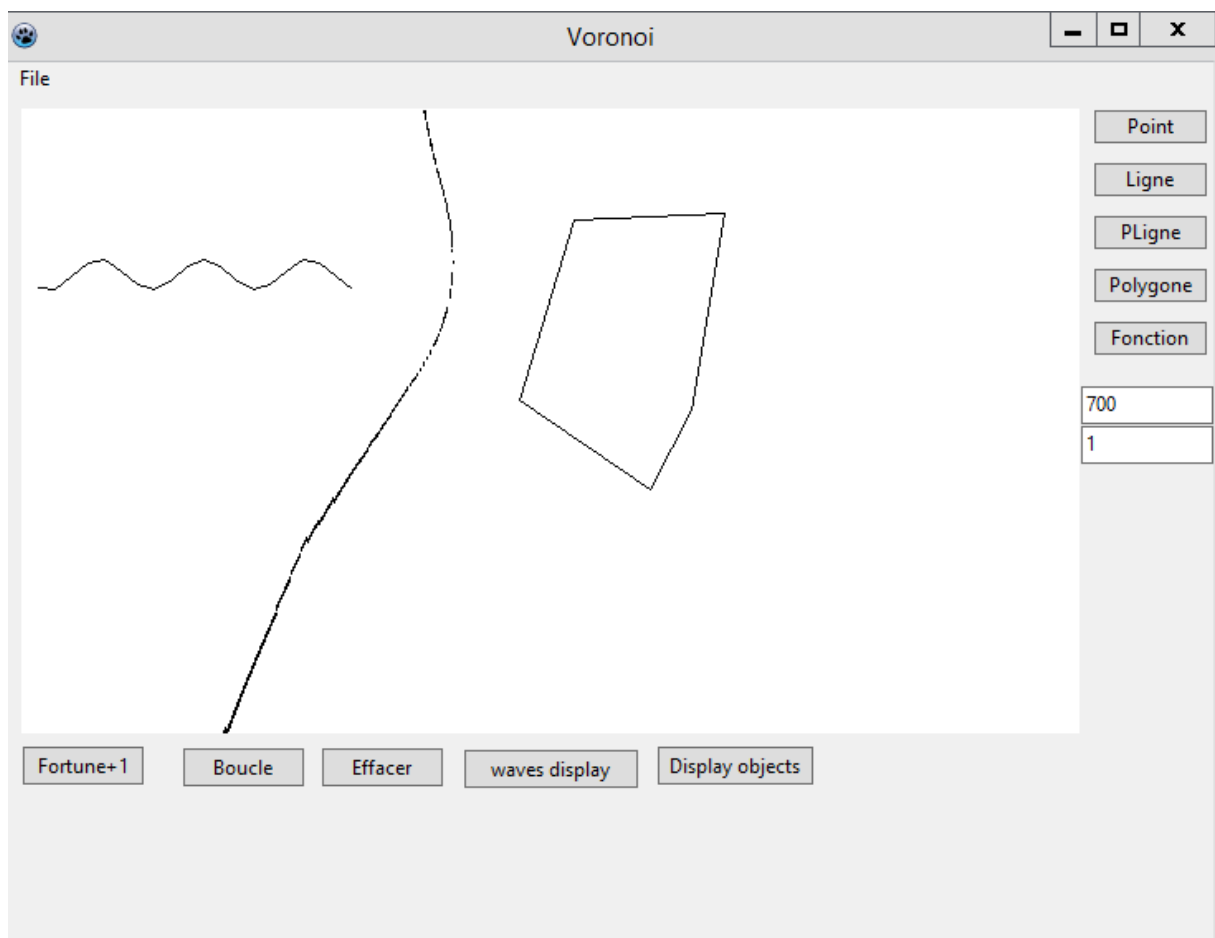
## 5.6 Détails sur le diagramme de Voronoi

Le diagramme de Voronoï fait partie du domaine de la géométrie algorithmique au même titre que la triangulation de Delaunay, l'enveloppe convexe, le rectangle englobant, etc. Il est utilisé dans de nombreux domaines dont la morphologie mathématique, la cartographie et la reconstruction de volume. Le diagramme de Voronoï consiste à diviser l'espace entre plusieurs sites formant ainsi des cellules de Voronoï. Les sites peuvent être des points, des segments, des objets etc. Il existe plusieurs algorithmes qui permettent le calcul du diagramme de Voronoi [33, 39, 79]. Parmi ces algorithmes se trouve l'algorithme de Steven Fortune [33]. Ce dernier a établi le principe pour les points, les segments (cependant, l'algorithme pour les segments ne fut pas implémenté) et les points pondérés. Nous présumons que le principe se prête assez bien pour les polygones, les polygones et les fonctions mathématiques. L'algorithme est implémenté sous *Lazarus*, nous effectuons différents tests sur des jeux de données de différents types et sur des cartes géographiques. Nous comparons les résultats avec un autre algorithme afin de vérifier la plausibilité de la théorie. Le choix s'est porté sur l'algorithme de Fortune en se basant sur le tableau 5.4.

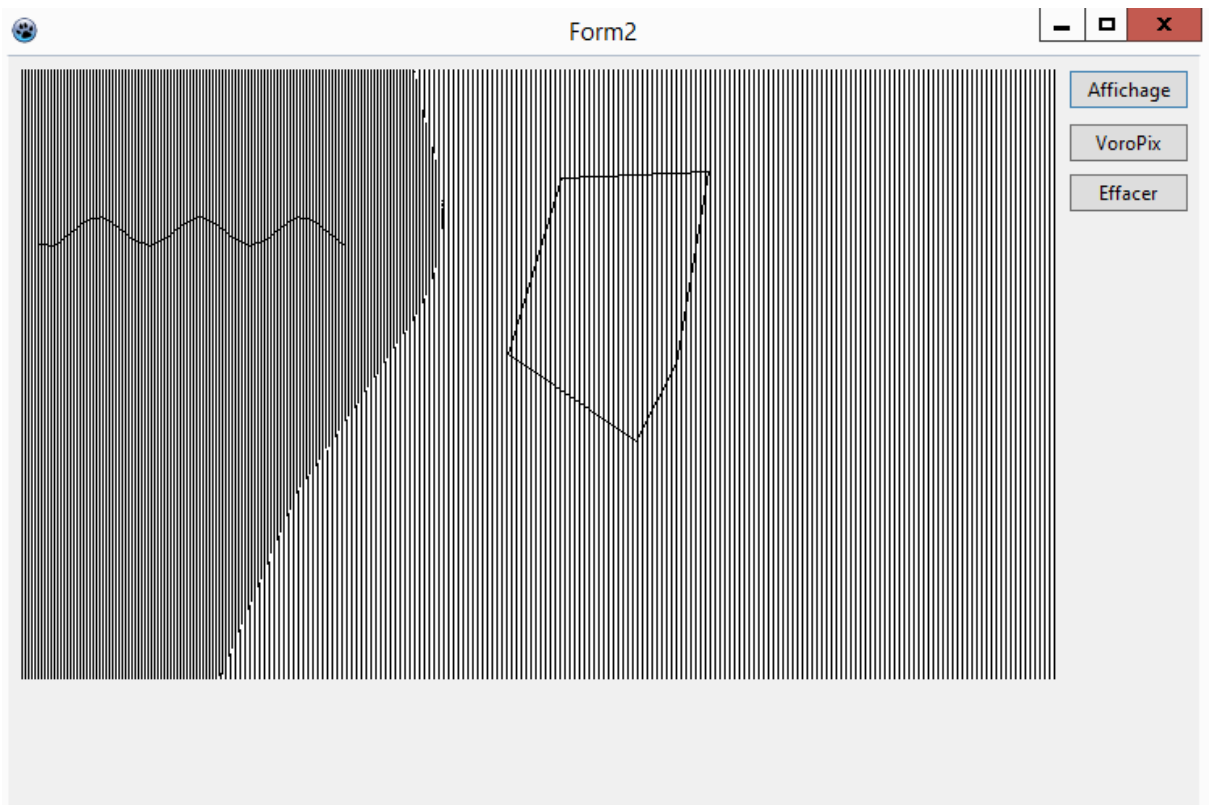
Tableau 5.4 Comparaison entre les algorithmes du calcul du diagramme de Voronoi [74]

	Intersection de demi-plans	L'algorithme de Green et Sibson[39]	L'algorithme de Shamos et Hoey[79]	L'algorithme de Steven Fortune[33]
La complexité	$O(n^2 \log n)$	$O(n)$	$O(n \log n)$	$O(n \log n)$
La simplicité	Simple	Simple	Très compliqué et difficile a implémenter	Simple
La généralisation	Il marche avec tous les Objets géométriques	Il marche seulement avec les points	Il marche avec tous les Objets géométriques	Il marche avec tous les Objets géométriques

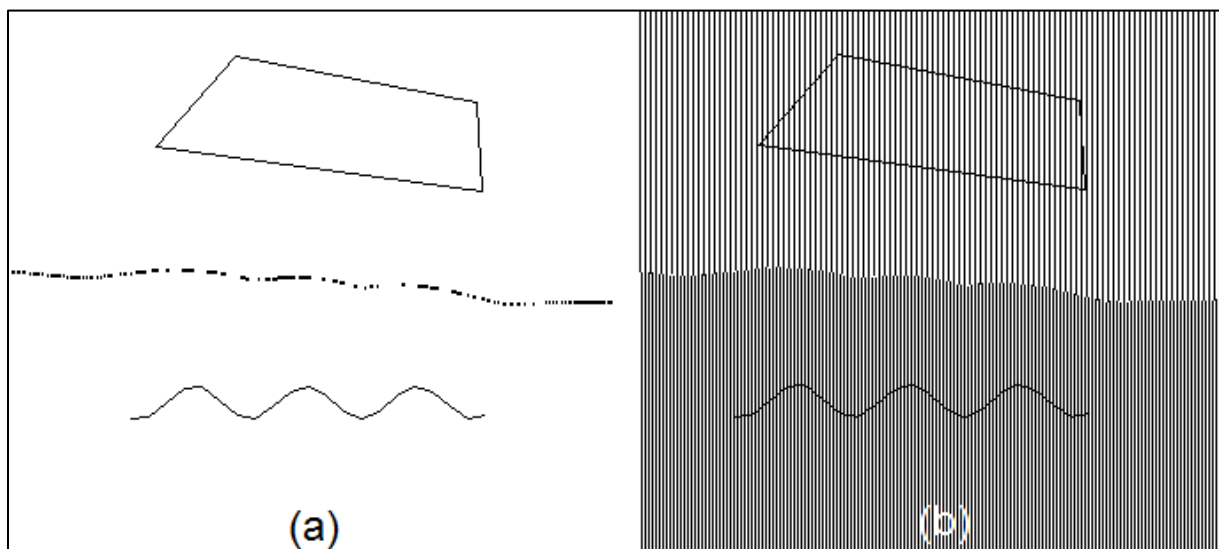
Le principe de l'algorithme de Fortune consiste à balayer le plan avec une ligne horizontale ou verticale et de former le diagramme au fur et à mesure, en formant des courbes de points du plan équidistants entre les sites (objets) rencontrés et la ligne de balayage, ensuite en intersectant ces courbes on obtient les points qui forment le graphe de Voronoi. On peut observer dans les figures Figure 5.12 et Figure 5.14(a) les résultats préliminaires ainsi que la comparaison avec un autre algorithme que nous avons choisi de nommer Voropix (Figure 5.13, Figure 5.14(b)). Les deux algorithmes sont encore en développement.



**Figure 5.12 Résultat 1 de Fortune sur une fonction sinus et un polygone**



**Figure 5.13 Résultat 1 de Voropix sur une fonction sinus et un polygone**



**Figure 5.14 Résultat 2 de Fortune (a) et de Voropix (b) sur une fonction sinus et un polygone**

Voropix a pu être testé sur les résultats d'autres travaux pour être validé comme on peut le voir dans les figures de Figure 5.15 à Figure 5.18.

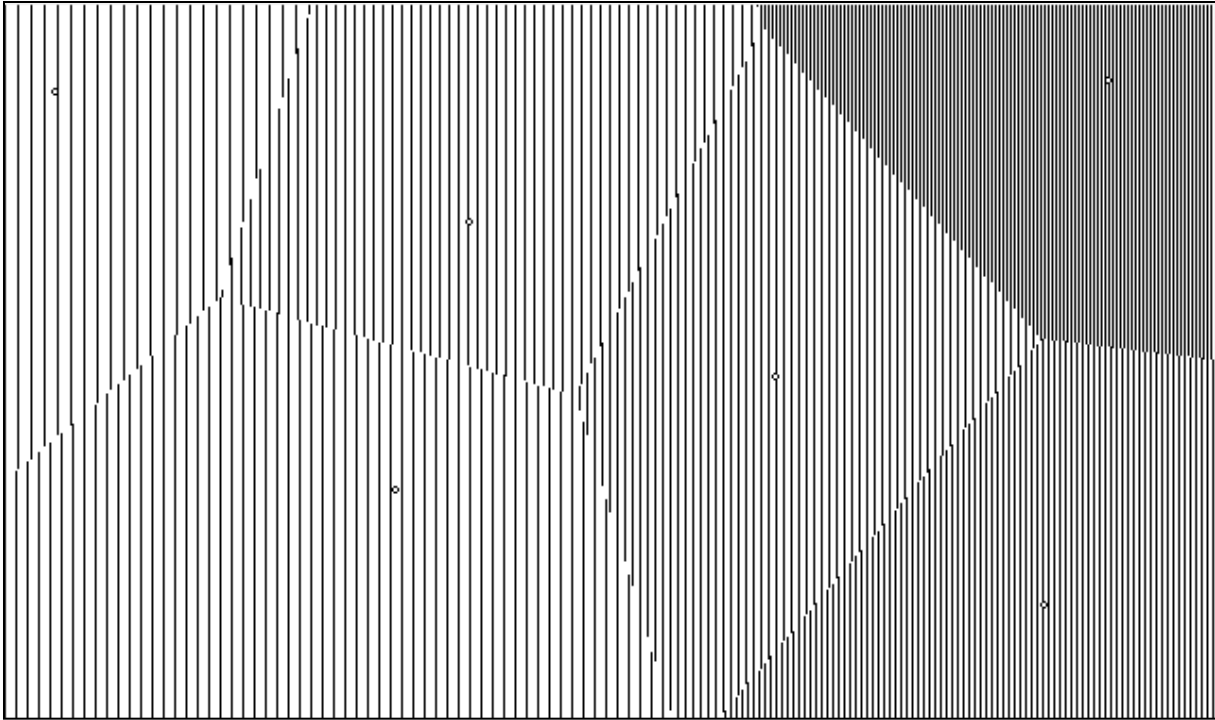


Figure 5.15 Voropix sur des points

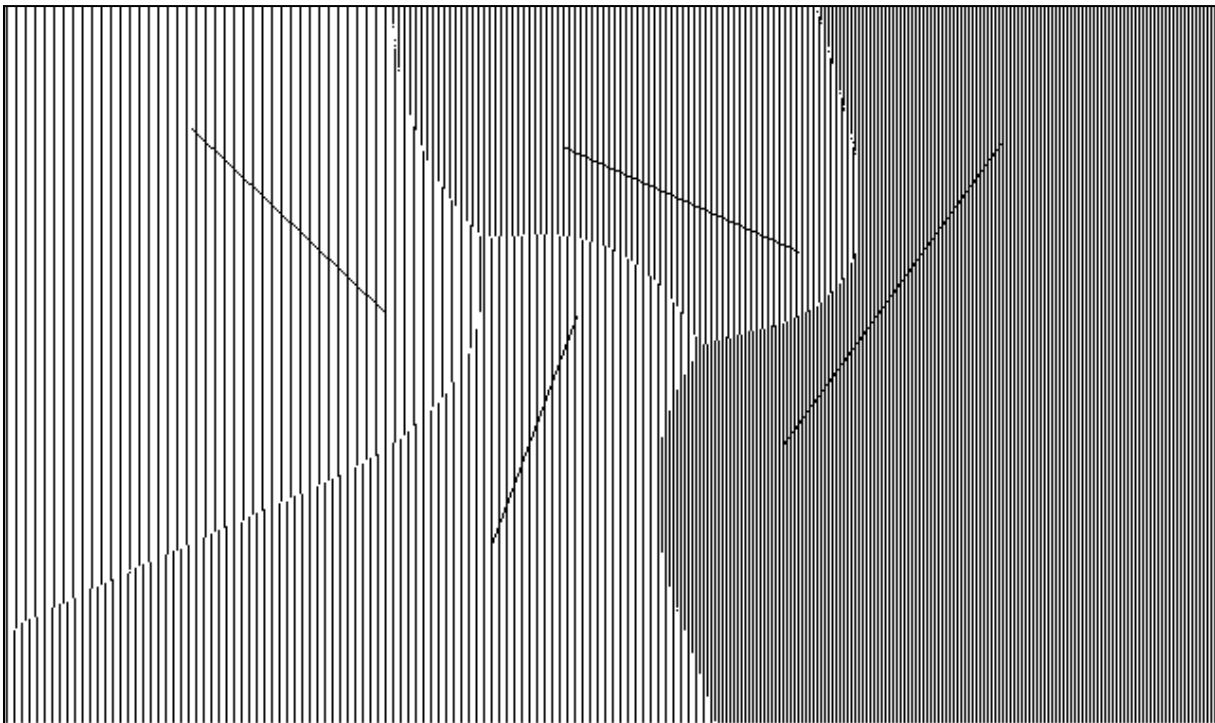


Figure 5.16 Voropix sur des lignes

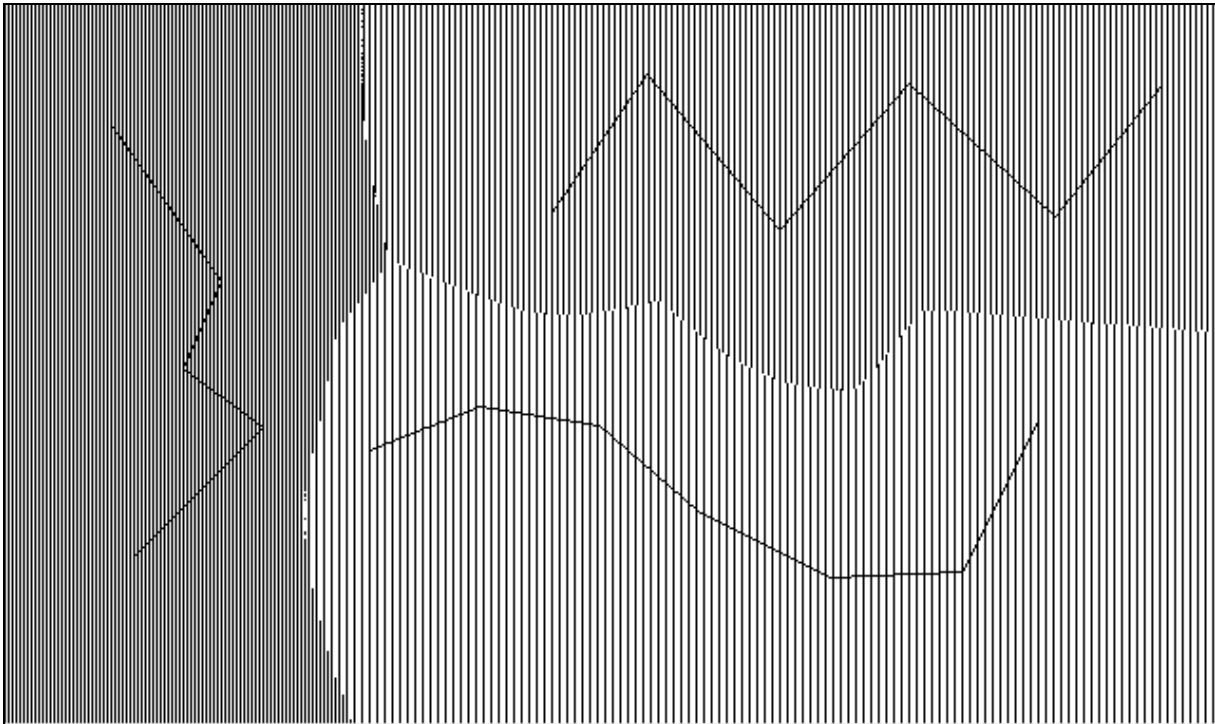


Figure 5.17 Voropix sur des polygones

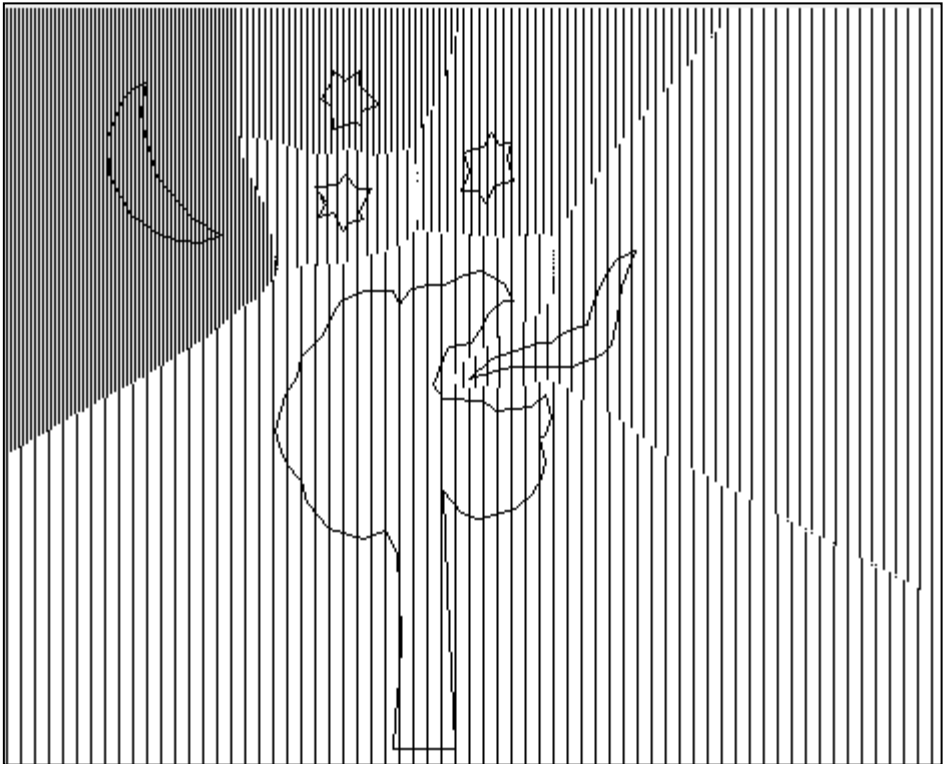
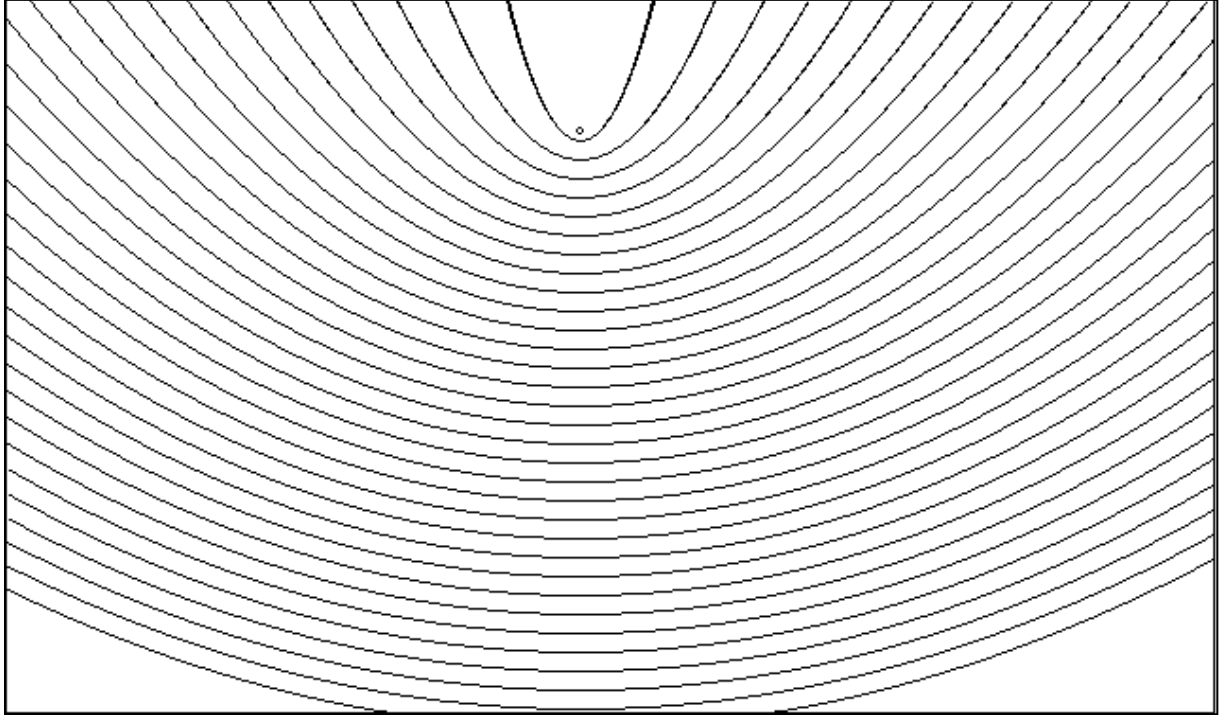
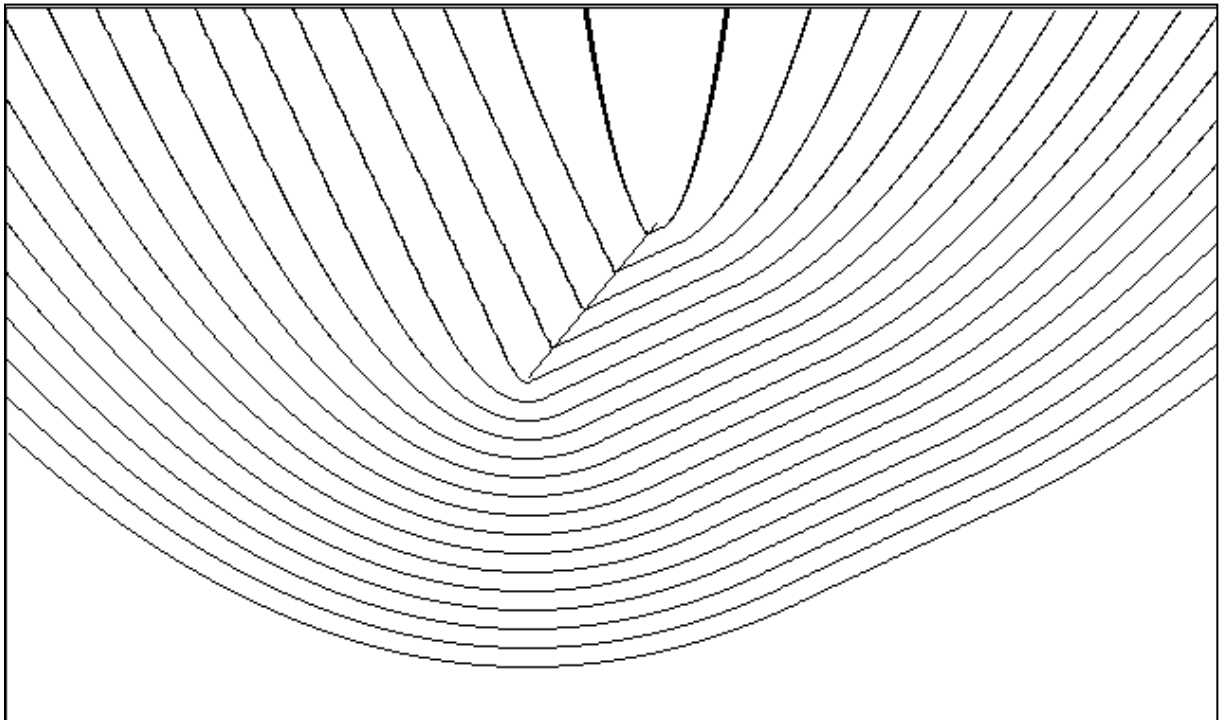


Figure 5.18 Voropix sur des polygones

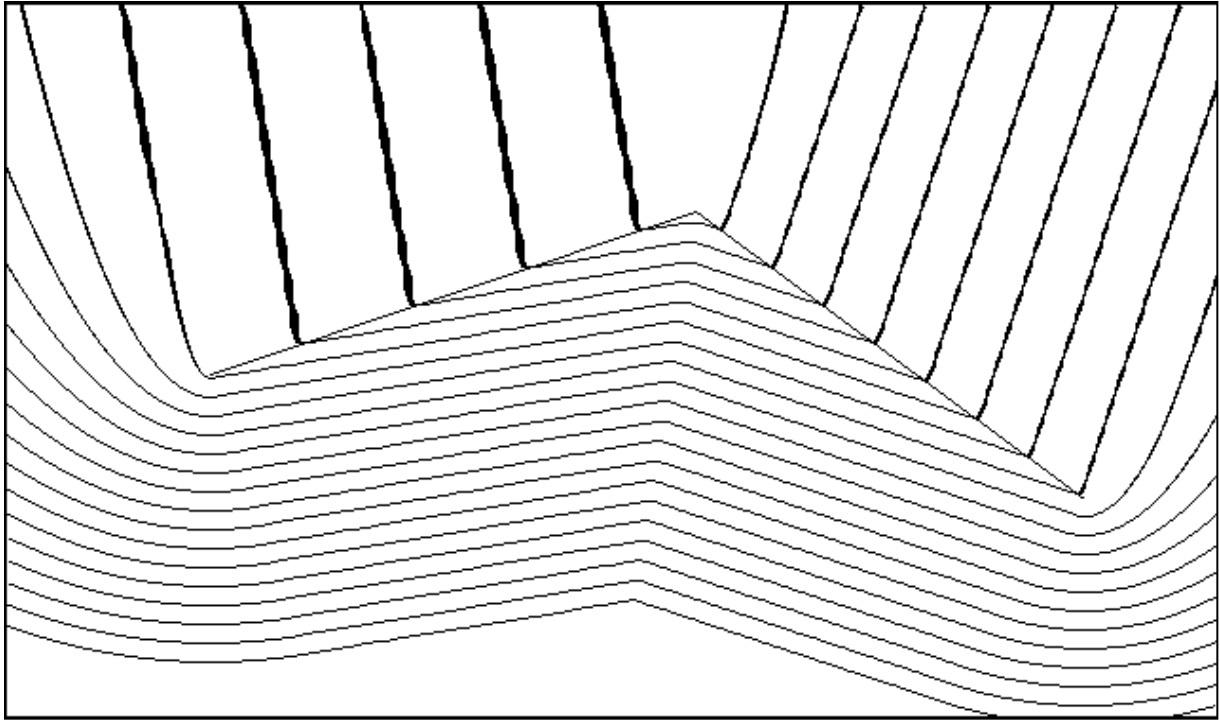
On peut observer dans les figures de Figure 5.19 à Figure 5.23 suivantes les courbes formées entre les objets de différents types et la ligne de balayage, pour différentes positions de celle-ci.



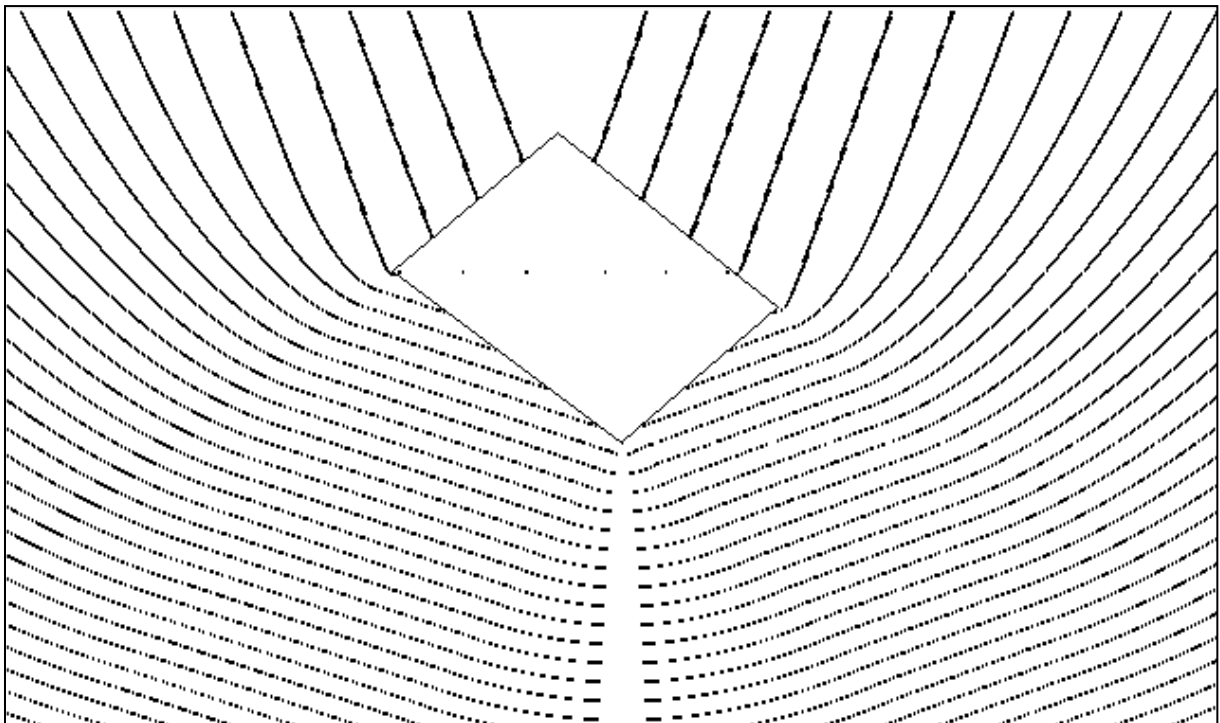
**Figure 5.19** Courbes formées entre un point et la ligne de balayage



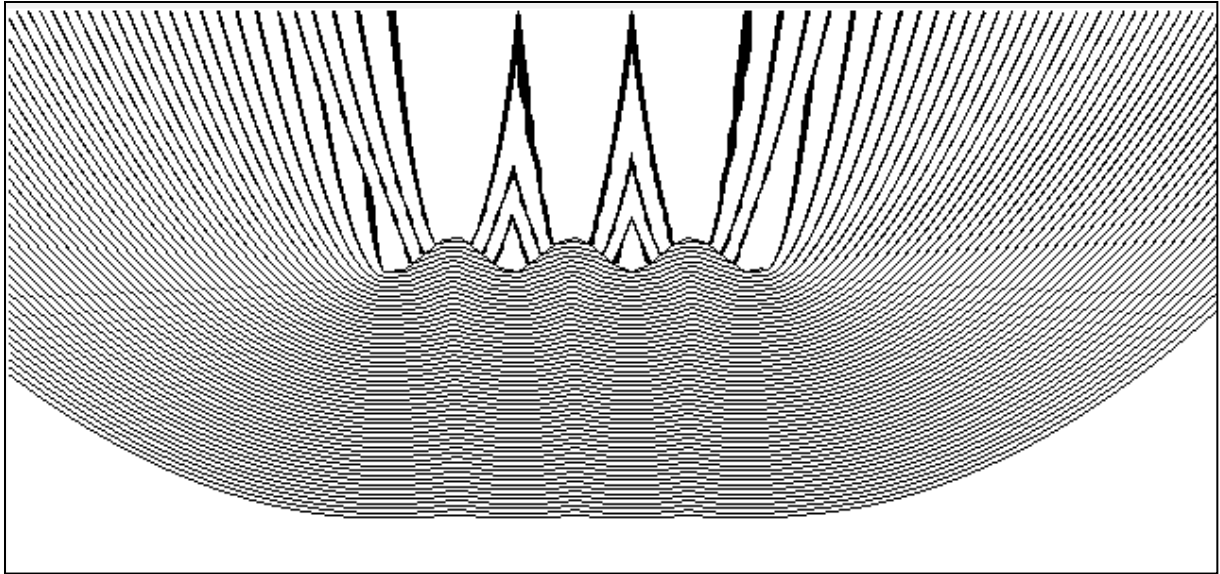
**Figure 5.20** Courbes formées entre une ligne et la ligne de balayage



**Figure 5.21** Courbes formées entre une polygones et la ligne de balayage

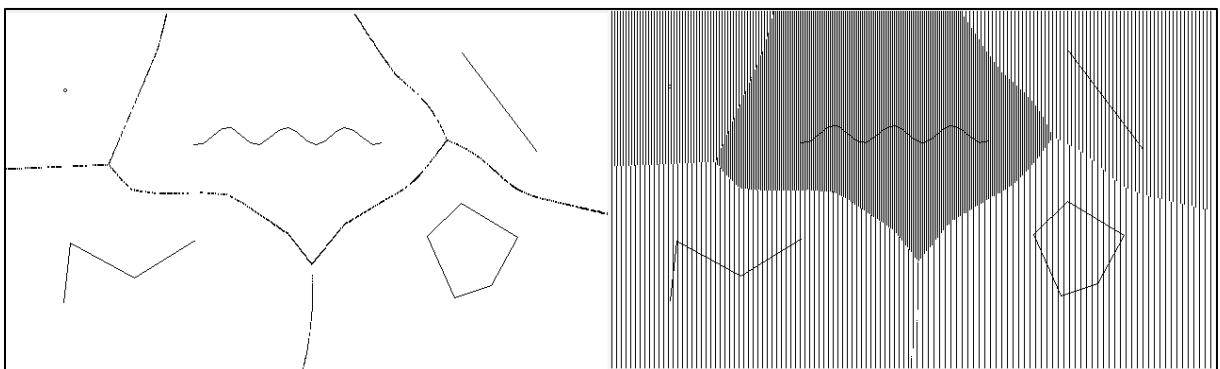


**Figure 5.22** Courbes formées entre un polygone et la ligne de balayage



**Figure 5.23** Courbes formées entre une sinus et la ligne de balayage

Enfin, nous avons dans la figure 5.24 les deux algorithmes Fortune et Voropix avec les différents types d'objets.



**Figure 5.24** Fortune à gauche et Voropix à droite avec les différents types d'objets

## 5.7 Synthèse et Conclusion

Dans cette section, nous avons défini un nouvel algorithme qui aide à résoudre le problème de la détection des motifs de forme quelconque et répétitive. L'algorithme a été appliqué à la zone de la ville de Dubaï et a pu détecter des modèles et leurs occurrences. L'algorithme donne de très bons résultats et a permis de détecter des occurrences de motif même si elles étaient orientées dans des directions différentes, en miroir ou en homothétie par

rapport au modèle du motif (à condition d'accepter un certain taux d'erreur). L'algorithme peut fonctionner avec d'autres algorithmes d'identification de motifs, tels que l'identification de motif linéaire et de grille pour une détection plus large. La méthode a été améliorée par la suite pour permettre de détecter plus de types de patterns ainsi que pour faciliter l'ajout de nouveaux modèles.

## Chapitre 6 : Proposition d'un algorithme de typification de patterns de patterns de bâtiments

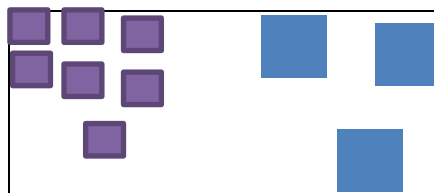
Ce chapitre est dédié à un nouvel algorithme de typification qui concerne lui aussi les patterns de bâtiments.

## 6.1.Introduction

Ce chapitre est consacré à notre algorithme de typification de patterns de bâtiments [42]. Nous allons donc présenter en détail l'algorithme ainsi que les tests et les résultats obtenus. Mais tout d'abord qu'est-ce que la typification, c'est ce qu'on va voir ci-dessous.

## 6.2.Définition de la typification

La typification de bâtiment (Figure 6.1) consiste à remplacer un groupe de bâtiments par un groupe réduit en nombre mais représentant la même emprise terrain et avec des bâtiments ayant les mêmes caractéristiques que les précédents. Cet algorithme est appliqué lorsque les bâtiments deviennent trop petits ou que la zone devient trop encombrée.



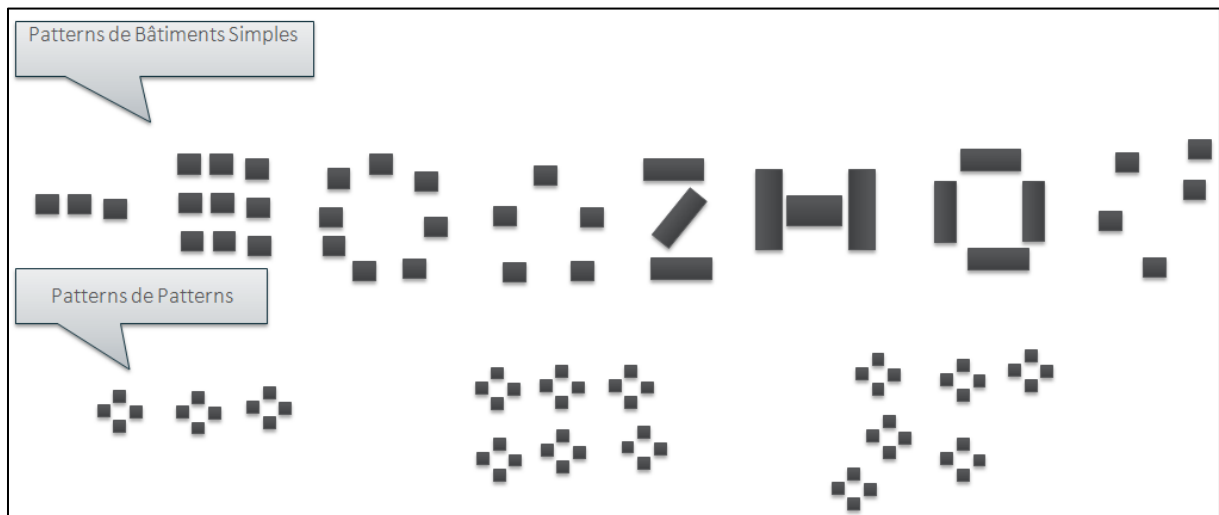
**Figure 6.1 Typification de bâtiments**

## 6.3.Problématique de la typification des patterns de bâtiments

Nous avons travaillé sur des cartes avec des échelles de 1: 3000 à 1: 50 000. On remarque que déjà à l'échelle 1: 3000, des problèmes de distance entre les bâtiments commencent à apparaître pour certaines zones. Ensuite, lorsque l'échelle diminue, nous commençons à avoir des problèmes d'illisibilité, d'imperceptibilité et de taille sur la carte. Nous nous sommes concentrés sur des échelles de 1: 25 000 à 1: 50 000. Dans des zones urbaines et rurales plus ou moins denses mais qui ne contiennent pas de bâtiments contigus, quel que soit leur type (pavillon, immeuble, organisation, etc.).

Les bâtiments peuvent avoir des positionnements particuliers (Figure 6.2) comme l'alignement, la grille, le cercle, en lettres d'alphabet ou simplement se faisant face de manière symétrique. Dans ce cas, nous parlons de patterns de bâtiments. Quand la structure est plus

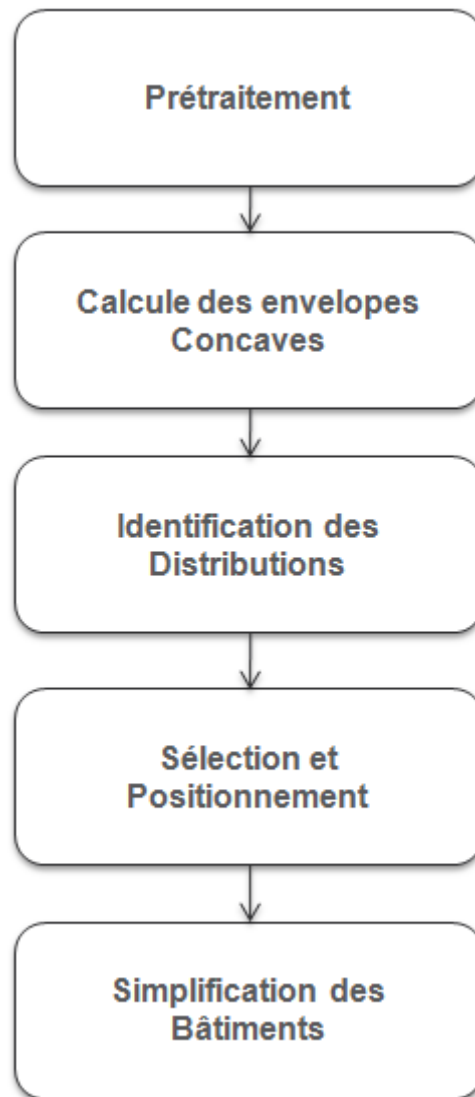
complexe quand on a des patterns positionnés en pattern on parle alors de pattern de patterns. Nous avons travaillé sur ce dernier cas mais notre méthode peut être appliquée facilement aux patterns simples.



**Figure 6.2 Exemples de patterns de bâtiments**

#### **6.4.La méthode**

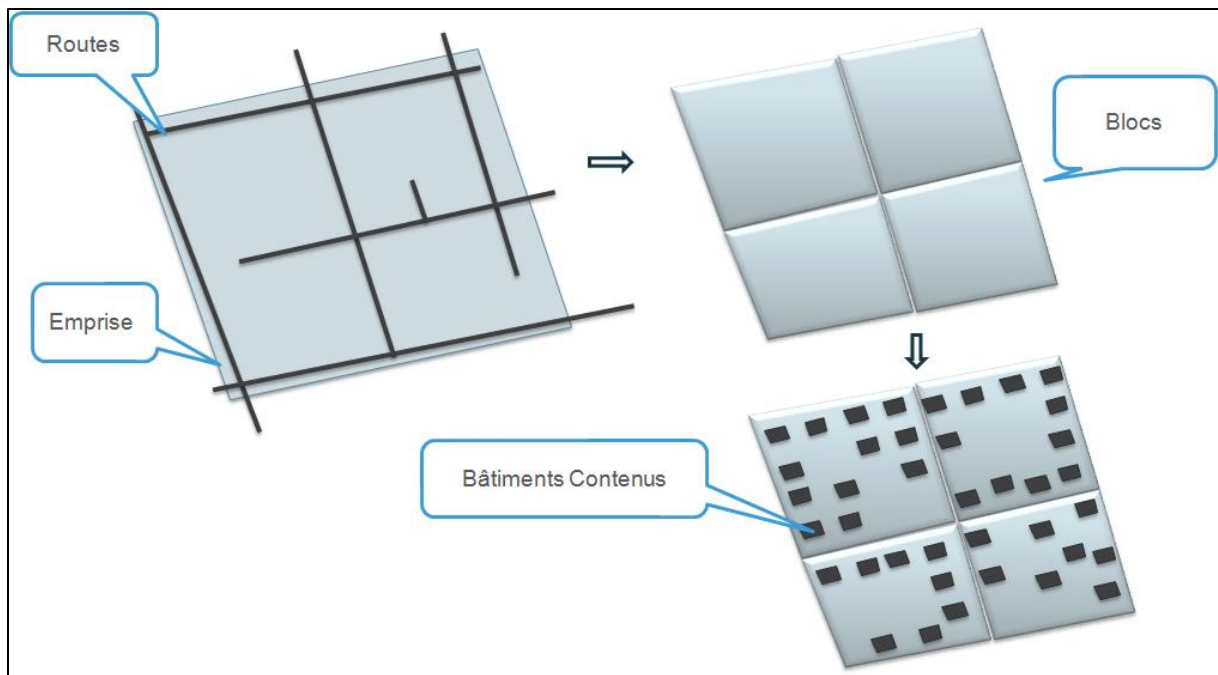
La méthode consiste en cinq étapes: le prétraitement qui nous permet de détecter les patterns de bâtiments, ensuite, nos occurrences de patterns sont agrégées avec des enveloppes concaves, ensuite, on procède à la détection de la distribution de nos enveloppes, puis nous sélectionnons et positionnons nos enveloppes sélectionnées en fonction de leur distribution, et enfin on simplifie les bâtiments. Les étapes de la méthode sont illustrées dans la figure 6.3 ci-dessous :



**Figure 6.3** Etapes de la méthode

#### **6.4.1. Prétraitement**

La méthode commence par le prétraitement. Un premier regroupement des bâtiments est effectué sur la base d'une division de la zone en blocs. Ensuite, nous attribuons à chaque bloc les bâtiments qu'il contient. La division en blocs se fait tout simplement en superposant les routes sur l'étendue de la zone d'étude (Figure 6.4).

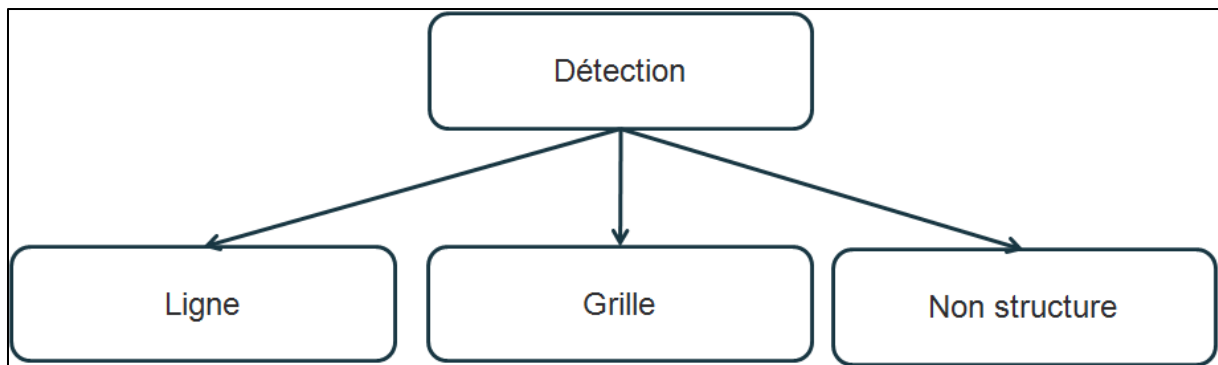


**Figure 6.4** Division d'une zone en blocs

Ensuite, nous avons une recherche de motifs répétés basée sur notre méthode décrite dans le chapitre précédent. Les occurrences de patterns sont remplacées par une forme enveloppante pour être traité comme un objet unique, on a choisi l'enveloppe concave, on aurait tout aussi bien pu choisir l'enveloppe convexe, une amalgamation, le rectangle englobant, le centre de gravité, etc.

#### **6.4.2. Détection de la dispersion**

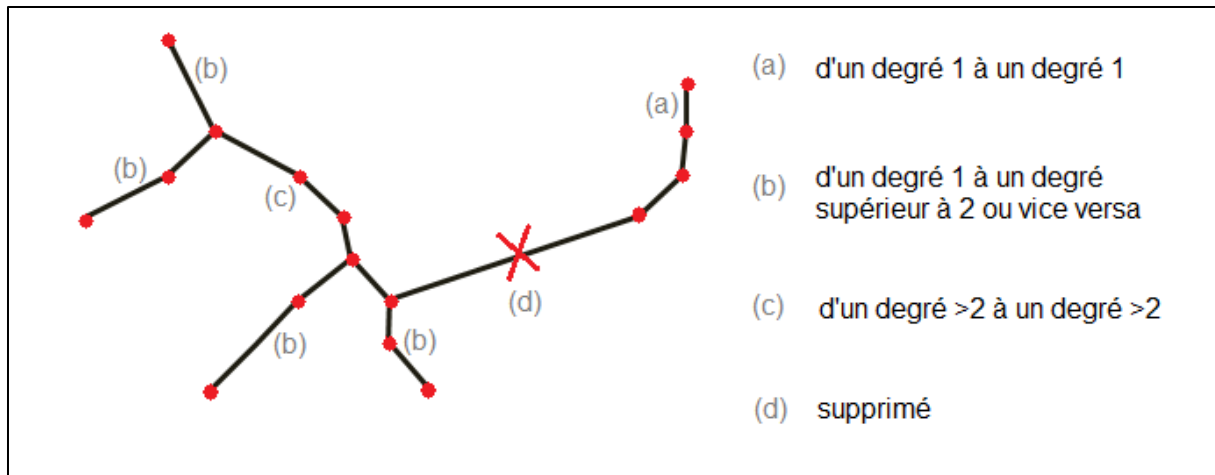
Par la suite, nous chercherons à connaître la dispersion de chaque pattern de patterns (Figure 6.5), s'il s'agit d'un alignement, d'une grille ou sinon c'est un motif non structuré. Ce sont les trois seuls patterns de patterns que nous avons pu voir dans les zones que nous avons étudiés.



**Figure 6.5 Détection des types de distributions spatiales des patterns de patterns**

#### **6.4.2.1. Détection de pattern de patterns linéaire**

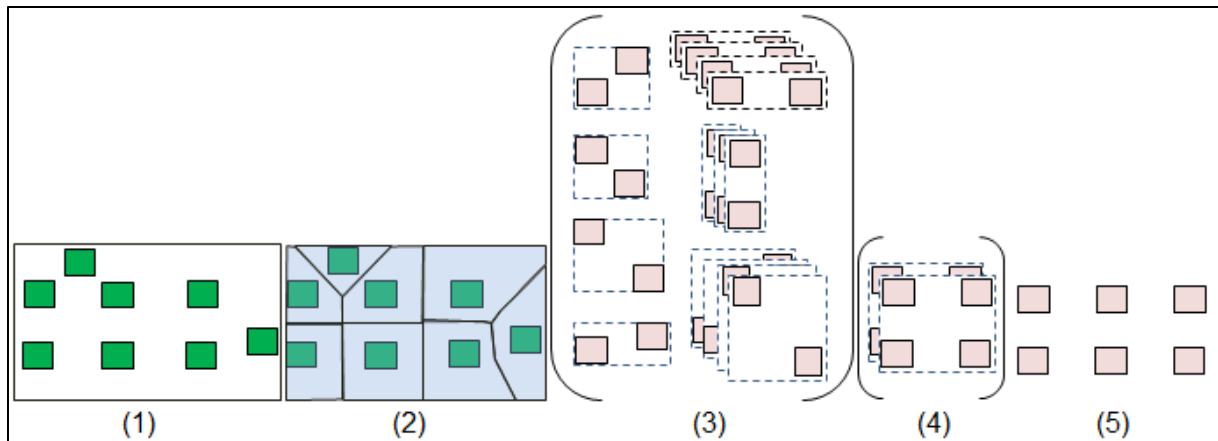
Pour détecter et isoler des groupes linéaires, nous calculons d'abord un arbre de recouvrement minimal des centres des éléments (enveloppes convexes ou bâtiments dans le cas de pattern simple), les arêtes de longueur supérieure à un certain seuil qui sont trop grandes sont supprimées. Ensuite, nous identifions trois types d'arêtes qui peuvent représenter un alignement. Une arête avec un sommet de départ et un sommet de fin participant à une seule arête (sommets avec un degré = 1). Une arête où le sommet de départ ne participe qu'à une arête (sommet avec un degré = 1) et le sommet final a plus de deux arêtes (sommet avec un degré > 2) ou vis versa. Une arête avec un sommet de début et de fin participant à plus de deux arêtes (sommet avec un degré > 2). Si une ligne a plus de deux sommets, elle est prise sinon elle est rejetée. L'algorithme commence bien sûr par rechercher un sommet participant à une seule ou plus de deux arêtes et finit par arriver à un sommet ayant soit une arête, soit plus de deux arêtes. Le cheminement se fait via des sommets participants à exactement deux arêtes. La méthode est illustrée dans la figure (Figure 6.6) ci-dessous :



**Figure 6.6** Extraction d'alignements à partir d'un arbre de recouvrement minimum de centres d'éléments. Les arêtes trop longues sont supprimées. (a) Une arête avec un point de départ et un point final participant à une seule arête. (b) Une arête où le point de départ participe à une seule arête et le point final a plus de deux arêtes. (c) Une arête avec des points de début et de fin participant tous deux à plus de deux arêtes.

#### 6.4.2.2. Détection de pattern de patterns en grille

Pour détecter et isoler une grille (Figure 6.7), nous procédons comme pour la détection de motifs répétés. C'est-à-dire qu'on va calculer le voisinage, par un diagramme de Voronoi, on va extraire  $V$ , le tableau binaire de voisinage, on va former itérativement des groupes de plus grande taille jusqu'à la taille 4. Ensuite, les groupes dont les centres ne forment pas un rectangle sont supprimés de  $V$ . Ensuite, on crée des matrices correspondant au groupes quadraires (un ordonnancement est nécessaire), avec comme valeurs de cellules les identifiants des objets. Ces matrices sont stockées dans le tableau  $W$ . Ensuite, on fusionne les matrices itérativement (quand elles ont une ou plusieurs lignes ou une ou plusieurs colonnes en commun) pour former des matrices de plus grandes tailles. Les redondances sont supprimées à chaque itération. On arrête quand  $W$  ne contient qu'une seule matrice. On forme un groupe d'objets correspondant à cette matrice grâce aux identifiants, ça sera notre grille.

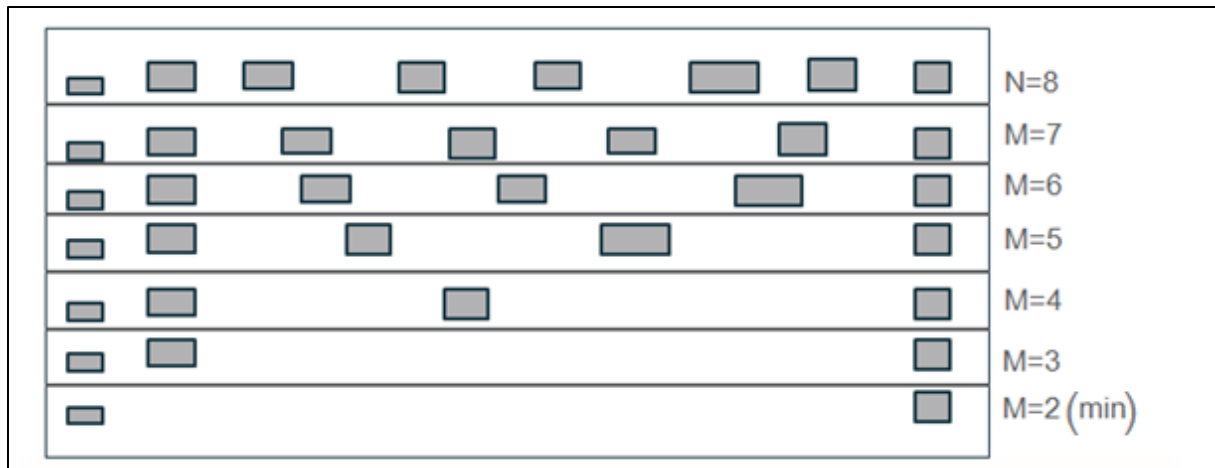


**Figure 6.7 Détection d'une distribution de grille, (1) est l'ensemble des objets d'entrée, (2) est un diagramme de Voronoi calculé sur ces objets, (3) est le vecteur de voisinage  $V$  à l'itération 1, (4) est le vecteur de voisinage  $V$  à l'itération 3, (5) est le résultat de la détection.**

L'ordre de détection est important, on commence par la détection des grilles, puis on effectue la détection des alignements.

#### 6.4.3. Sélection et positionnement

Pour l'alignement, nous nous sommes inspirés des travaux de [45] décrits dans le chapitre 3 (§3.7.1). Soit  $n$  le nombre d'objets dans le groupe d'origine, et  $m$  le nombre d'objets cibles, nous calculons ensuite  $m$  positions, cela nous donne la dispersion des éléments futurs. Les éléments les plus proches de ces emplacements restent et y sont placés, les autres sont éliminés. L'image suivante (Figure 6.8) donne une idée de la distribution donnée par la formule avec différentes valeurs de  $m$ :



**Figure 6.8 Distribution de la formule de l'alignement avec différentes valeurs de  $m$**

Pour la Grille (Figure 6.9): la suppression se fait en fonction des dimensions de la forme de la grille (on extrait ainsi l'orientation et l'allongement du rectangle englobant orienté), la suppression se fait par ligne entière et colonne entière. Soit  $n$  le nombre de colonnes de la grille d'origine,  $m$  le nombre de colonnes de la grille résultantes,  $a$  la largeur de la grille,  $es$  est le total des espaces d'une colonne,  $c$  la largeur d'un élément,  $e$  la distance entre deux éléments d'une même ligne,  $r$  le rapport entre les valeurs  $c$  et  $e$ ,  $e'$  la distance entre deux éléments résultants d'une même ligne,  $c'$  la largeur d'un élément résultant, cette valeur est connue. Le calcul de  $m$  est donné par l'ensemble d'équations 6.1 (le calcul du nombre de lignes suit la même procédure).

$$es = a - (n \times c)$$

$$e = \frac{es}{n - 1}$$

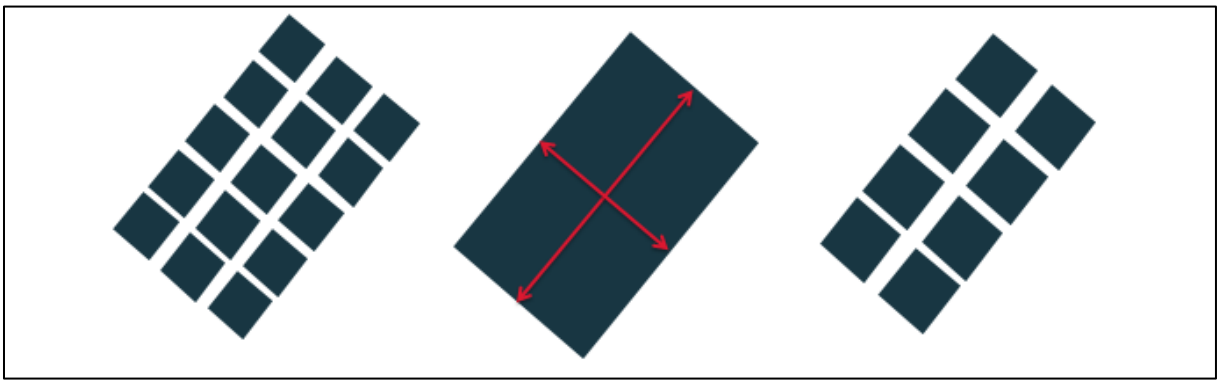
$$r = \frac{e}{c}$$

$$e' = r \times c'$$

(6.1)

$$a = m \times c' + (m - 1) \times e'$$

$$m = \frac{(a + e')}{(c' + e')}$$



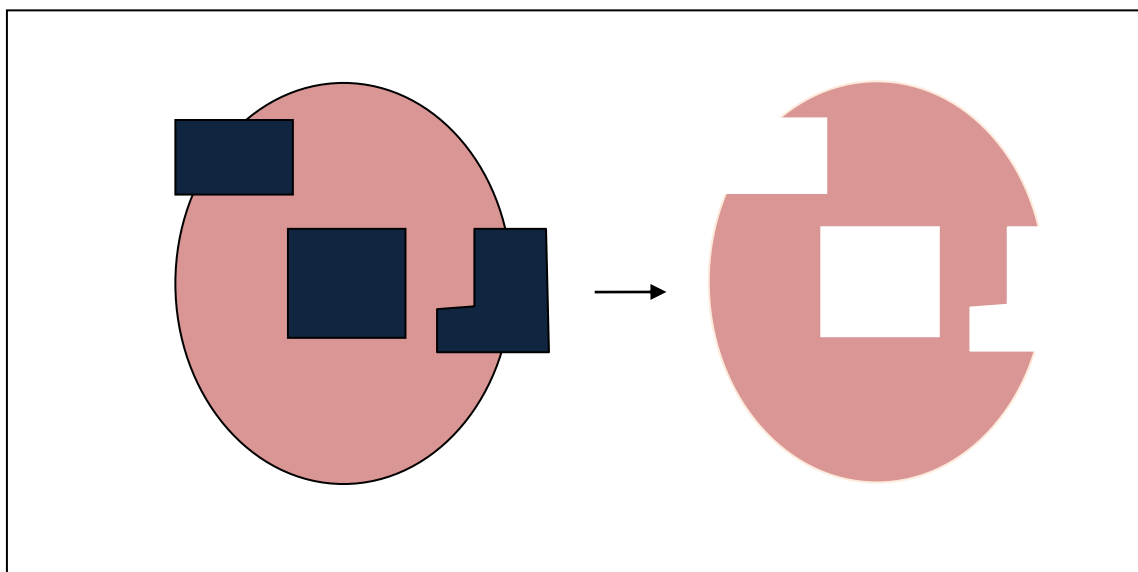
**Figure 6.9** Typification d'un pattern grille

Pour le motif non structuré, on effectue une recherche de voisinage basée sur une triangulation de Delaunay. Le nombre d'éléments à garder à l'échelle cible est calculé comme suit: soit  $s$  la surface de l'enveloppe convexe d'un élément avec sa taille d'origine, et soit  $s'$  sa surface après traitement et amplification,  $n$  le nombre de bâtiments d'origine et  $m$  le nombre d'éléments à l'échelle cible (équation 6.2).

$$m = \left(\frac{s}{s'}\right) \times n \quad (6.2)$$

Ensuite, pour la sélection nous nous appuyons sur une fonction de coût basée sur trois critères : l'importance, jugée sur la taille, données par la surface. Cette valeur est divisée par la surface maximale pour ramener la valeur à une plage de 0 à 1. Pour réduire le cout des bâtiments importants ce rapport de surface est soustrait de 1. La deuxième valeur est le

nombre de voisins à une certaine distance *dist* paramétrable. Cette valeur est aussi divisée par la valeur maximale. La dernière valeur est la zone d'encombrement (Figure 6.10), elle consiste à soustraire les surfaces des voisins proches ainsi que celle de l'objet lui-même de la surface d'un cercle de rayon *dist*. La surface qui reste représente la valeur d'encombrement, plus elle est petite plus la zone est encombrée et plus elle est grande plus la zone est désencombrée. Pour ramener la valeur à une valeur comprise entre 0 et 1, on divise par la valeur maximale et pour augmenter le coût d'un objet encombré, ce rapport est soustrait de 1.

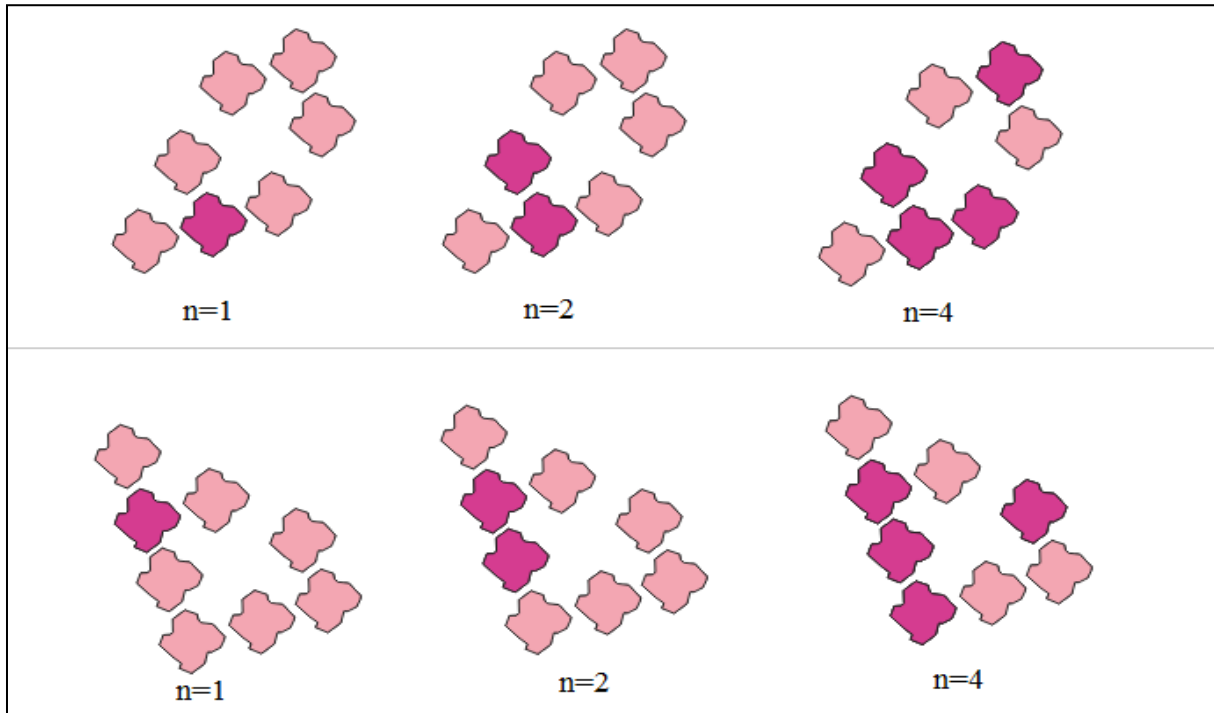


**Figure 6.10 Zone d'encombrement**

Les trois valeurs sont multipliées par des poids. Ces poids par défaut sont égaux à 1/3 afin de donner la même importance aux trois critères. Si l'on veut privilégier un critère en particulier, il suffit d'augmenter son poids et si l'on veut éliminer un critère, il suffit de mettre son poids à 0. Soit **coût<sub>i</sub>** le coût de l'objet numéro **i**, **surf<sub>i</sub>** la surface de l'objet **i**, **maxsurf** la surface maximale (surface du plus grand objet du groupe), **nv<sub>i</sub>** le nombre de voisins dans la zone d'encombrement de l'objet **i**, **maxnv** la valeur maximale de **nv**, **ze<sub>i</sub>** la surface de la zone d'encombrement de l'objet **i**, **maxze** la valeur maximale de **ze**. Soient **p1**, **p2**, **p3** les poids de nos trois critères d'évaluation. La fonction de coût est donnée par l'équation 6.3.

$$coût_i = p_1 \times \left(1 - \frac{surf_i}{maxsurf}\right) + p_2 \times \left(\frac{nv_i}{maxnv}\right) + p_3 \times \left(1 - \frac{ze_i}{maxze}\right) \quad (6.3)$$

On peut voir sur la figure suivante (Figure 6.11) le résultat de la fonction de coût sur des groupes en fonction du nombre d'objets qu'on veut supprimer :

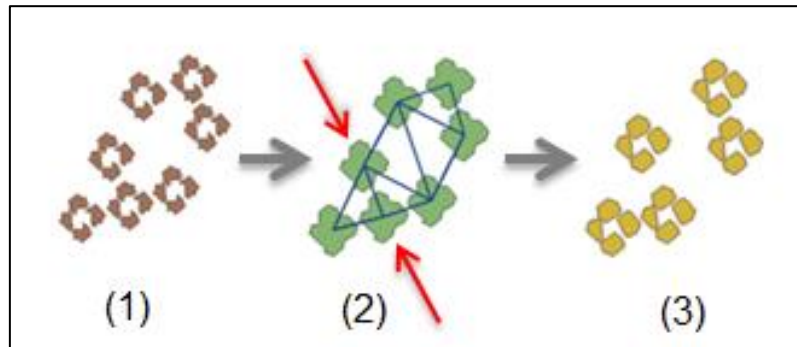


**Figure 6.11 Résultat de la fonction de coût**

A la place d'une fonction de distribution, nous nous appuyons sur un algorithme de déplacement pour calculer les nouveaux emplacements des objets.

- Pour le déplacement, Les objets sont déplacés comme suit:
  - Si un objet n'a pas de voisin éliminé, sa position reste la même.
  - Si un objet a des voisins éliminés, sa position est alors une moyenne entre les positions de ses voisins éliminés et sa propre position.
- La position calculée est considérée comme un emplacement idéal et l'objet n'y est pas directement placé, avant, il faut s'assurer qu'il n'y a pas de risque de provocation d'un conflit de superposition. Le déplacement est alors effectué progressivement (avec un pas

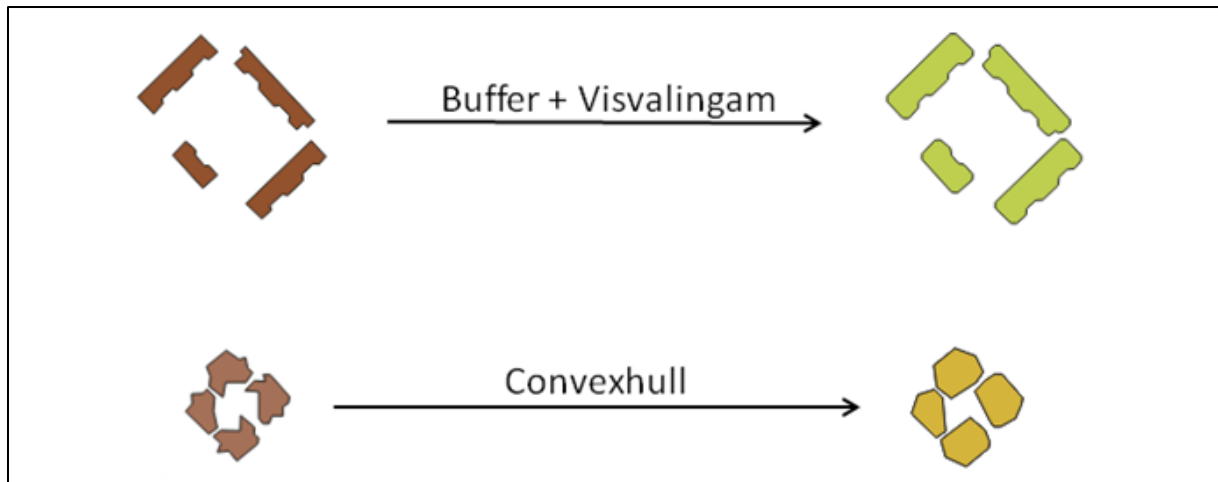
choisit, déterminé et de petite valeur) en contrôlant si le déplacement ne provoque pas de superposition. Si le déplacement d'un objet à un moment donné risque de provoquer un conflit de superposition, son déplacement est alors stoppé. La méthode est illustrée dans la figure 6.11.



**Figure 6.12** Sur cette image, nous avons en (1) le groupe de motifs répétés en entrée, en (2) nous calculons des enveloppes concaves et une triangulation de Delaunay sur ses enveloppes, les flèches rouges indiquent des éléments avec un coût plus élevé que les autres, en (3) nous avons le résultat de l'algorithme après avoir supprimé les éléments à coût élevé et déplacé les éléments restants.

#### 6.4.4. Simplification des bâtiments

On peut traiter tous les bâtiments de la même manière sans prendre en compte le modèle du motif, dans ce cas on peut simplifier les bâtiments en utilisant au besoin: une amplification de la taille (buffer), un remplacement par enveloppe convexe, un remplacement par le rectangle englobant orienté, la simplification du contour avec l'algorithme de Douglas et Peuckerou celui de Visvalingam (Figure 6.13).

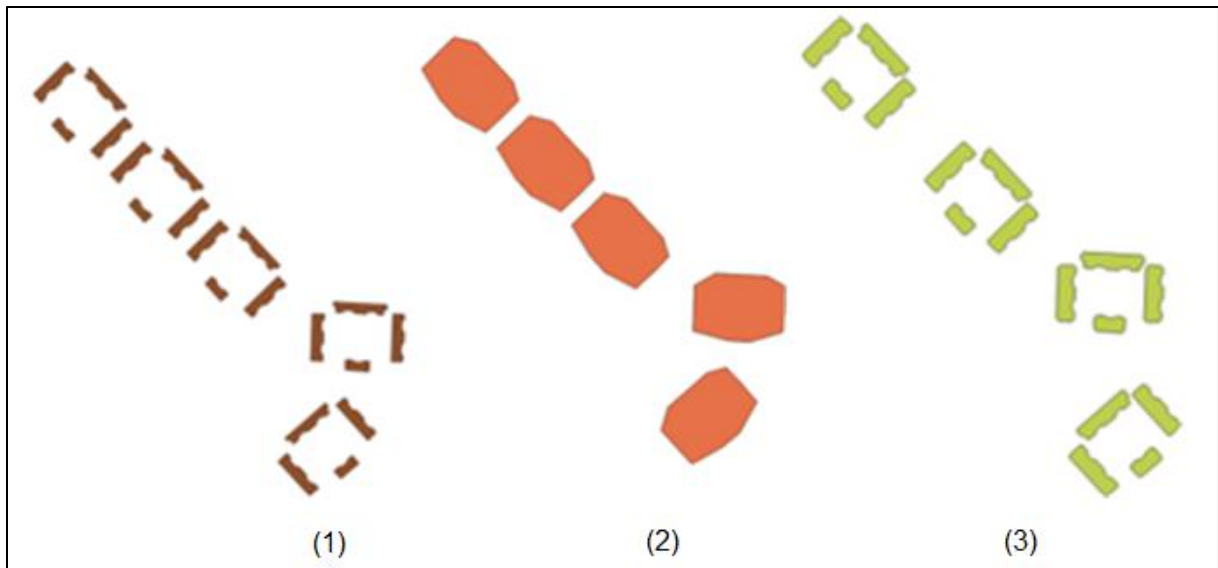


**Figure 6.13 Simplification du bâtiment**

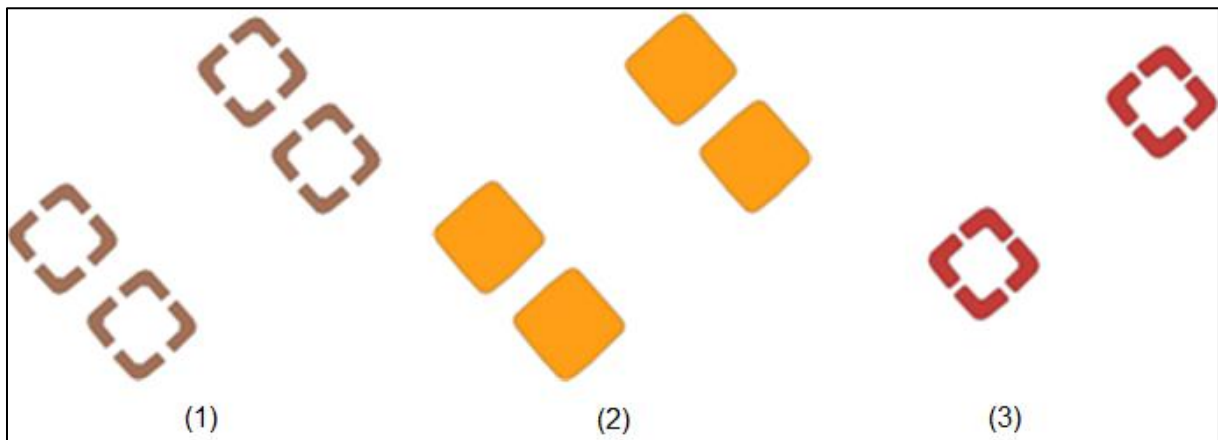
Il est également possible de généraliser le modèle en appliquant par exemple une fusion ou agrégation d'occurrences. Sinon, on peut typifier le modèle lui-même.

## 6.5. Résultats

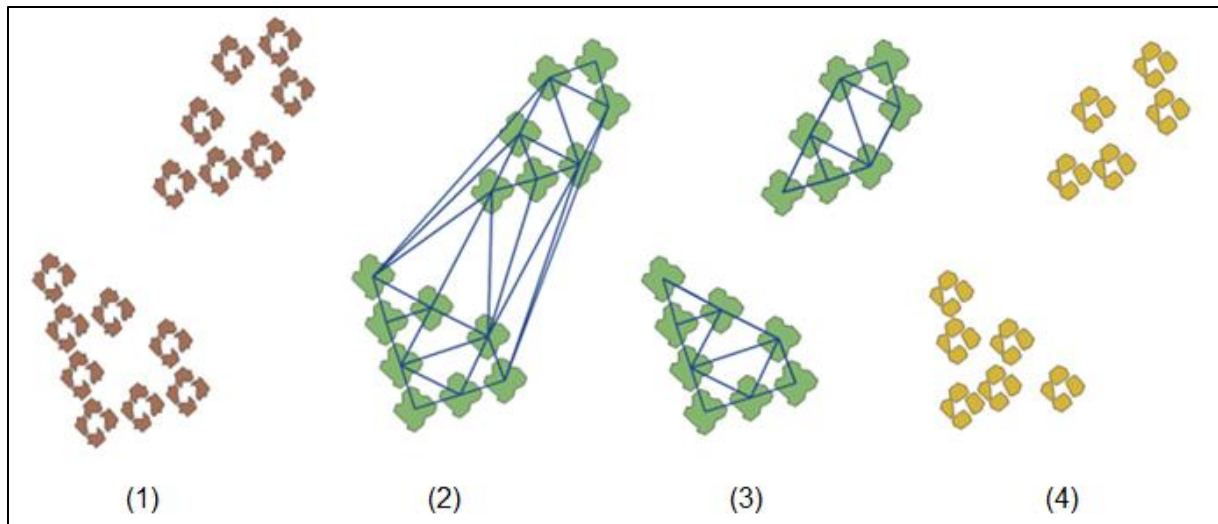
La méthode a été testée sur des échantillons des villes de Laghouat et Blida en Algérie. Les données proviennent d'OpenStreetMap et de GoogleMaps. La méthode a été implémentée sous QGIS qui est un logiciel de cartographie open source, elle prend en charge PyQGIS qui est basé sur le langage python augmenté par les bibliothèques spatiales QGIS. Nous présentons ici quelques-uns de nos résultats. Nous pouvons voir sur la figure 6.14 l'application de la méthode sur un motif de motifs linéaire. Sur la figure 6.15, nous avons l'application de la méthode sur un motif de motifs en grille et sur la figure 6.16, nous avons l'application de la méthode sur un motif de motifs non structurés.



**Figure 6.14** Application de la méthode sur un motif linéaire de motifs. L'image (1) représente les bâtiments d'origine, l'image (2) représente la coque convexe des motifs répétés détectés et l'image (3) représente le résultat final de la typification.

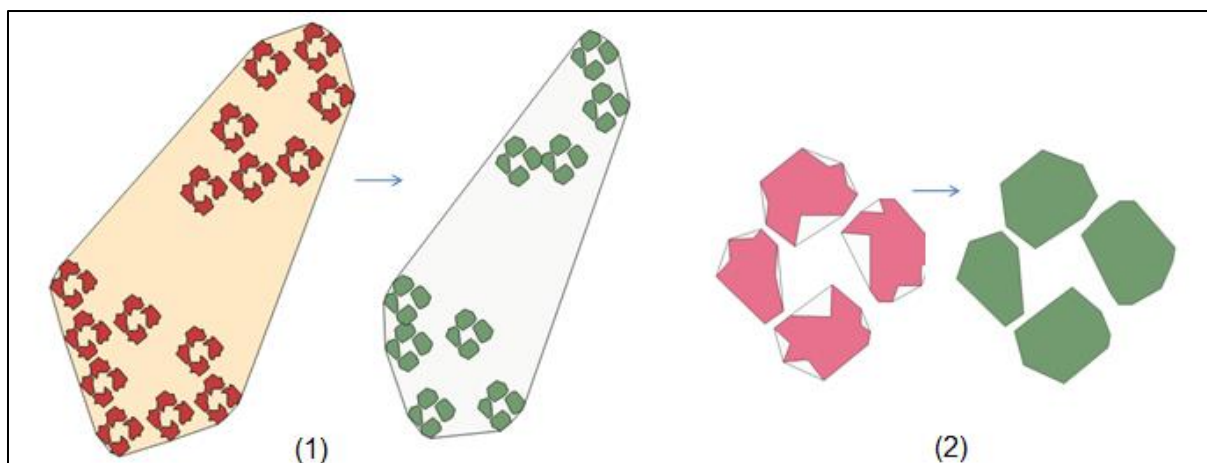


**Figure 6.15** Application de la méthode sur un motif de motifs en grille. L'image (1) représente les bâtiments d'origine, l'image (2) représente l'enveloppe convexe des motifs répétés détectés et l'image (3) représente le résultat final de la typification.



**Figure 6.16 Application de la méthode sur un motif de motifs non structuré. L'image (1) représente les bâtiments d'origine, l'image (2) représente l'enveloppe convexe des motifs répétés détectés, une triangulation de Delaunay lui a été appliquée, sur l'image (3) les segments trop longs (supérieurs à un seuil paramétrable) ont été supprimés et l'image (4) représente le résultat final de la typification.**

Nous remarquons visuellement que les motifs répétés en sortie sont similaires aux originaux et sont reconnaissables. Il y a moins de bâtiments mais on a presque la même impression de population. Les motifs de motifs sont identifiables et la forme linéaire, grille et non structurée a été respectée. Pour évaluer le résultat, nous nous sommes appuyés sur la préservation de l'étendue du terrain et la forme des bâtiments (Figure 6.16).



**Figure 6.17** La figure (1) représente la mesure de l'étendue du terrain donnée par le rapport des surfaces de l'enveloppe convexe de tous les bâtiments avant et après l'application de la méthode. L'image (2) représente la mesure des formes des bâtiments donnée en comparant le rapport de la surface du bâtiment à son enveloppe convexe avant et après l'application de la méthode.

Les valeurs entre les groupes d'origine et les groupes résultants ont donc été comparées (tableau 6.1). L'étendue du terrain est donnée par l'aire de l'enveloppe convexe totale des bâtiments. Nous calculons ensuite le rapport entre les surfaces des enveloppes des bâtiments du groupe d'origine et celles des groupes résultants. Comme ce rapport se rapproche de 1, les valeurs sont donc proches les unes des autres, l'étendue est alors conservée. La forme des bâtiments est obtenue par le rapport de la surface d'un bâtiment et de son enveloppe convexe, nous calculons ce rapport pour les bâtiments d'origine et les bâtiments résultants nous remarquons que la valeur de ce rapport entre les deux est sensiblement la même. Les résultats obtenus sur les autres échantillons sont similaires.

**Tableau 6.1** Mesure du résultat pour les trois types de dispersion avec les champs : (N) Nombre d'occurrences du motif répété ou (L|C) le nombre de lignes et le nombre de colonnes, (M) nombre d'occurrences du motif répété résultant ou (L|C) nombre de lignes et nombre de colonnes, (Etendue) l'étendue du terrain, (Forme O) la forme des bâtiments d'origine et (Forme R) la forme des bâtiments résultants.

Dispersion	N / L C	M / L C	Etendue	Forme O	Forme R
Linéaire	5	4	1,05	0,88	0,93
Grille	2   2	1   2	0,94	0,78	0,71
Non structuré	15	11	1,13	0,83	1

La typification est testée seule, il serait intéressant de la tester dans un processus de généralisation avec des algorithmes tels que la fusion, l'agrégation, la couverture, etc.

## **6.6.Conclusion**

La typification des bâtiments est une opération de généralisation. Nous avons vu qu'il y avait différents types de groupes de bâtiments appelés patterns et que chacun nécessitait une typification particulière. Les résultats obtenus sont intéressants. Cependant, la plupart des outils ne sont pas disponibles dans les outils SIG accessibles, l'implémentation a occupé une grande partie du projet et nous avons donc dû limiter l'application à certains modèles. Pour les applications futures, nous avons l'intention d'ajouter d'autres types de motifs (cercle, étoile, Z, H, E...) et éventuellement l'introduction d'une méthode de reconnaissance de formes.

Dans les trois chapitres précédents nous avons proposé et testé trois nouveaux outils de généralisation qui ont permis de répondre à certains besoins de la généralisation dans le domaine des routes et des bâtiments. Dans la prochaine partie de ce document, nous présentons nos conclusions et nos perspectives.

## Conclusion générale

Cette partie de ce document consiste en une conclusion générale suivie de nos perspectives dans le domaine de la généralisation cartographique automatique.

## 7.1. Conclusion

Nous avons vu dans ce document une présentation générale sur le domaine du spatial et de la production des cartes. Nous avons vu que l'une des opérations principales de réalisation de carte est la généralisation cartographique. C'est une tâche complexe qui consiste à dériver des cartes à petites échelles à partir de cartes à plus grandes échelles. Elle est d'habitude réalisée par des cartographes experts en généralisation. Mais le processus étant couteux et de longue haleine plusieurs travaux d'automatisation ont été tentés. Trois approches se sont distinguées à savoir, la multi-représentation, la généralisation automatique et la méthode combinée.

La généralisation automatique est un domaine jeune. Une solution complète n'existe pas encore sur le marché. Nous avons vu que parmi les raisons, il y avait le manque de mesures et d'algorithmes de généralisation. Pour répondre à ce manque nous avons pu proposer et tester trois nouveaux outils de généralisation :

- Un nouvel algorithme de généralisation des séries de virages.
- Une nouvelle mesure de généralisation pour la détection de patterns de bâtiments répétitifs.
- Un nouvel algorithme de généralisation pour la typification de patterns de bâtiments.
- Un quatrième travail étant en cours sur le diagramme de Voronoi de polygones, de polygones et de fonctions. Le diagramme de Voronoi étant un algorithme géométrique utilisé dans plusieurs domaines. Cependant, on peut retrouver le diagramme de Voronoi de points facilement dans les outils traditionnels (ArcGIS, MapInfo, FME, QGIS, etc.), le diagramme de Voronoi de segments plus rarement mais les diagrammes pour les polygones, les polygones et les fonctions sont quasiment inexistantes dans ces outils.

Tout d'abord dans le domaine du traitement des routes et plus précisément les séries de virages. Nous avons vu qu'une route pouvait être découpée en plusieurs parties pour un meilleur traitement. Les parties linéaires (sans virages), les virages isolés et les séries de virages. Nous avons vu que ces dernières étaient sujettes à l'empâtement routier, c'est-à-dire lorsque des parties de la même route fusionnent entre elles, du fait de la symbolisation. Nous avons vu aussi que les algorithmes existants ne permettaient pas de résoudre l'empâtement sur tous les types de séries de virages mais plutôt sur les séries de virages simples. Nous avons donc proposé un algorithme qui permet de corriger l'empâtement sur les types de séries qui ne sont pas prises en charge par les algorithmes existants et qui permet de préserver l'emprise terrain du résultat.

Nous avons vu ensuite, dans le domaine de la généralisation des bâtiments, qu'il y avait certaines dispositions particulières des bâtiments qu'on appelle patterns de bâtiments. Ces derniers pouvaient être disposés linéairement, en grille, sous forme de lettres d'alphabet, en formes géométriques ou encore sous forme quelconque mais répétitivement comme le motif d'un tissu. Si les méthodes existantes permettent de détecter des patterns de formes connues, elles ne permettent pas la détection des patterns quelconques. Nous avons donc proposé une nouvelle méthode pour les détecter basée sur un ensemble d'opérations géométriques à savoir la translation, la rotation, la miroirisation et l'homothétie. La méthode a pu isoler les patterns du reste des bâtiments ainsi que des uns par rapport aux autres, elle a aussi permis de détecter des patterns même orientés, en miroir ou en homothétie les uns par rapport aux autres.

En fin, nous avons pu voir l'opération de la typification et plus précisément la typification de pattern de bâtiments. Nous avons vu que le problème de cette opération est la variété des patterns de bâtiments, de plus, l'existence de patterns de patterns. Chaque type nécessitant un principe de détection et de typification particulier. Nous avons pu proposer une

méthode de typification qui permette de détecter et de typifier les patterns de patterns (qui peut s'appliquer aux patterns simples) disposés linéairement, en grille ou de manière non structurée. La méthode a pu donner de bons résultats sur les échantillons testés.

## **7.2.Perspectives**

Au niveau des domaines que nous avons traités, nous envisageons des travaux d'études sur les types de séries de virages et de ce fait une proposition de plus d'outils pour les traiter.

Au niveau des patterns de bâtiments, nous envisageons une méthode de reconnaissance de formes pour détecter plus de patterns de bâtiments simples pour ensuite, les typifier. Au niveau des patterns de patterns de bâtiments, il serait intéressant d'adapter les autres algorithmes de généralisation (autre que le typification) à leur traitement, comme le déplacement, la fusion, l'agrégation, etc.

Il n'existe pas de solution de généralisation complète. Le domaine reste dans le besoin de plus d'algorithmes et de mesures de généralisation et de plus de travaux sur les processus de généralisation.

Et pour faciliter et encourager les travaux dans le domaine, il serait utile d'implémenter les outils de généralisation mis à jour par les travaux dans le domaine dans les logiciels de cartographie les plus répandus. Ceci faciliterait leur étude, comparaison ou utilisation dans un autre projet de généralisation.

# Bibliographie

# Bibliographie

- [1]. Abbas I (1994) Base de données vectorielles et erreur cartographique: problèmes posés par le contrôle ponctuel; une méthode alternative fondée sur la distance de Hausdorff: le contrôle linéaire. Thèse de doctorat. Université de Paris-Diderot, Paris.
- [2]. Abam MA (2010) Streaming algorithms for line simplification. *Discrete and Computational Geometry* 43(3): 497-515.
- [3]. AGENT Consortium. (1999) Selection of basic measures. Rapport DD2. Disponible sur: <http://agent.ign.fr/deliverable/DC1.html>.
- [4]. Alertes-meteo.com. Géographie. <https://alertes-meteo.com/geographie/teledetection.php>. Accédé en Mai 2021
- [5]. Alessandro C (2003) Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping. Thèse de doctorat. Université de Zurich, Zurich.
- [6]. Anders KH (2006) Grid Typification. *Progress in Spatial Data Handling*. [https://doi.org/10.1007/3-540-35589-8\\_40](https://doi.org/10.1007/3-540-35589-8_40)
- [7]. Baella B, Pla M, Palomar-Vasquez J, Pardo-Pascual J (2007) Spot heights generalization : deriving the relief of the topographic database of catalonia at 1 :25 000 from the master database. Workshop sur la généralisation et la représentation multiple, Russie.
- [8]. Balboa JLG, Lopez FJA (2005a) Road line segmentation for cartographic generalization: a wavelet based procedure. 22<sup>ième</sup> (ICC) International Cartographic Conference, Espagne.
- [9]. Balboa JLG, Lopez FJA, Luque RL (2005b) Road line classification for cartographic generalization : a neural net approach. 22<sup>ième</sup> (ICC) International Cartographic Conference, Espagne.
- [10]. Balboa JLG, Lopez FJA (2008) Generalization-oriented Road Line Classification by Means of an Artificial Neural Network. *Geoinformatica*. 12: 289–312 <https://doi.org/10.1007/s10707-007-0026-z>
- [11]. Balboa JLG, Lopez FJA (2009) Sinuosity pattern recognition of road features for segmentation purposes in cartographic generalization. *Pattern Recognition*. 42(9): 2150-2159.

- [12]. Baldin R (2010) Context awareness and typification in building generalisation. Thèse de doctorat. Université de Padua, Italy.
- [13]. Bédard Y. (1998-2012) Generalization and Multiple Representations for on Demand Map Production and Delivery. Projet GEMURE pour le réseau GEOIDE (GEOmatics for Informed DEcisions).
- [14]. Bernier E (2002) Utilisation de la représentation multiple comme support à la génération de vues de bases de données géospatiales dans un contexte SOLAP. Mémoire de Maîtrise géomatique. Université Laval. Canada.
- [15]. Bobzien M, Burghardt D, Petzold I, Neun M, Weibel R (2008) Multi-representation Databases with Explicitly Modeled Horizontal, Vertical, and Update Relations. *Cartography and Geographic Information Science*, 35:1, 3-16, DOI: 10.1559/152304008783475698
- [16]. Brassel K, Weibel R (1988) A review and conceptual framework of automated map generalization. *International Journal of Geographical Information Systems*. 2(3):229-244.
- [17]. Burghardt D, Cecconi A (2007) Mesh simplification for building typification. *International Journal of Geographical Information Science*. 21(3)  
<https://doi.org/10.1080/13658810600912323>
- [18]. Battenfield BP, McMaster RB (1991) Constraints on rule formation. *Map Generalization : Making Rules for Knowledge Representation*. Longman Scientific & Technical, Harlow, United kingdom.
- [19]. Cebrykow P (2017) Cartographic generalization yesterday and today. *Polish Cartographical Review*. 49(1): 5-15. doi: <https://doi.org/10.1515/pcr-2017-0001>
- [20]. Cecconi A., Weibel R. and Barrault M (2002) Improving automated generalisation for on demand Web mapping by multiscale database. *Proceeding du Symposium sur la théorie géospatiale, Processing and Applications*, Ottawa, Canada.
- [21]. Cetinkaya S, Basaraner M, Burghardt D (2014) Proximity-based grouping of buildings in urban blocks: a comparison of four algorithms. *Geocarto International*. 30(6):618-632  
DOI:10.1080/10106049.2014.925002
- [22]. Chaudhry O, Mackaness WA (2008) Automatic identification of urban settlement boundaries for multiple representation databases. *Computers, Environment and Urban Systems*. 32(2):95- 109.

- [23]. Cosma I.M. (2004). Modèle de données pour la production cartographique sur le Web : mise en œuvre des représentations multiples en GML. Maîtrise d'informatique. Université Laval. Canada.
- [24]. Damour S. (1994). Generalisation du batis : La simplification des contours. Rapport de stage de D.E.A. des Sciences de l'Information Géographique. Institut Géographique National.
- [25]. Deng H, Fang W, Yang O, Li Y (2003) A model of road network generalisation based on genetic algorithms. International Cartographic Conference, Afrique du Sud.
- [26]. Deng M, Liu Q, Cheng T, Shi Y (2011) An adaptive spatial clustering algorithm based on Delaunay triangulation. *Comput Environ Urban Syst.* 35:320–332.
- [27]. Deng M, Tang J, Liu Q, Wu F (2017) Recognizing building groups for generalization: a comparative study. *Cartography and Geographic Information Science.* 45(3):187-204 DOI: 10.1080/15230406.2017.1302821
- [28]. Douglas D, Peucker T (1973) Algorithms for the reduction of the number of points required to represent a digitised line or its caricature. *The Canadian Cartographer.* 10: 112-122.
- [29]. Duchene C (2004) Généralisation Cartographique par Agents Communicants : le modèle CartACom. Thèse de doctorat, Sorbonne Université, Paris.
- [30]. Du S, Shu M, Feng C (2015) Representation and discovery of building patterns: a three-level relational approach. *International Journal of Geographical Information Science.* 30(6):1161-1186 DOI: 10.1080/13658816.2015.1108421
- [31]. Du S, Luo L, Cao K, Shu M (2016) Extracting building patterns with multilevel graph partition and building grouping. *Journal of Photogrammetry and Remote Sensing.* 122:81-96.
- [32]. Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. International Conference on Knowledge Discovery and Data Mining, États Unis.
- [33]. Fortune, S. (1987) A sweepline algorithm for Voronoi diagrams. *Algorithmica* 2, 153. <https://doi.org/10.1007/BF01840357>

- [34]. Fritsch E. (1997) Représentation de la géométrie et des contraintes cartographiques pour la généralisation du linéaire routier. Thèse de doctorat. Université de Gustave Eiffel, Paris.
- [35]. Gaffuri J (2008) Généralisation automatique pour la prise en compte de thèmes champ: le modèle GAEL. Thèse de doctorat. Université de Gustave Eiffel, Paris.
- [36]. Gao P, Liu Z, Han F, Tang L, Xie M (2015) Accelerating the computation of multi-scale visual curvature for simplifying a large set of polylines with Hadoop. *GIScience & Remote Sensing*. 52(3): 315-331 DOI: [10.1080/15481603.2015.1035528](https://doi.org/10.1080/15481603.2015.1035528)
- [37]. Goethem AV, Meulemans W, Speckmann B, Wood J (2014) Exploring Curved Schematization. *IEEE Pacific Visualization Symposium*, Yokohama. doi: 10.1109/PacificVis.
- [38]. Gong X, Wu F (2016) A typification method for linear pattern in urban building generalisation. *Geocarto International*. 33(2):189-207.
- [39]. Green PJ, Sibson R, (1978) Computing Dirichlet Tessellations in the Plane, *The Computer Journal*, Volume 21, Issue 2, May 1978, Pages 168 173  
<https://doi.org/10.1093/comjnl/21.2.168>
- [40]. Hamini N. (2008) Résolution des conflits cartographiques dans un entrepôt de données spatiales par un nouveau système de généralisation basé sur une approche combinée. Mémoire de magistère. Ecole Supérieure d'Informatique. Algérie.
- [41]. Hamini, N., (2019) Building pattern identification for map generalization Proceedings de la 2<sup>ème</sup> Conference Springer de Arabian Journal of Geosciences (CAJG-2), Tunisia.
- [42]. Hamini, N., Yagoubi, M.B. (2020) Building pattern typification Proceedings de la 3<sup>ème</sup> Conference Springer de Arabian Journal of Geosciences (CAJG-3), Tunisia.
- [43]. Hamini, N., Yagoubi, M.B. (2021) The MinimumBreak algorithm applied to a series of road turns. *Arab J Geosci* **14**, 140. <https://doi.org/10.1007/s12517-021-06471-2>
- [44]. Hamini, N., Yagoubi, M.B. (2021) Fortune's algorithm for calculating the Voronoi diagram of polylines, polygons and functions. Proceedings du premier meeting annuel Springer de Mediterranean Geosciences Union (MedGU-1), Turquie.
- [45]. Hangouet JF (1998) Apprentissage supervisé pour la généralisation cartographique : application en bord de ville. Thèse de doctorat université de Gustave Eiffel, Paris.
- [46]. ICA. International Cartographic Association. <https://icaci.org/>. Accédé en Mai 2021.

- [47]. Jabeur N and Sahli N (2010) Progressive and autonomous approach towards the generation of personalized web maps. 10th International Conference on Intelligent Systems Design and Applications. 1264-1269. doi: 10.1109/ISDA.2010.5687082.
- [48]. Jenks GF (1989) Geographic logic in line generalisation. *Cartographica*. 26(1): 27-42.
- [49]. Karypis G, Han EH, Kumar V (1999) CHAMELEON: Hierarchical clustering using dynamic modeling. *Computer*. 32:68–75.
- [50]. Lang T (1969) Rules for robot draughtsmen. *Geographical Magazine*. 42: 50-51.
- [51]. Lecordix F, Plazanet C, Lagrange JP (1997) A Platform for Research in Generalization: Application to Caricature. *GeoInformatica*. 1(2): 161-182.
- [52]. Legrand C, Duchene C, Lecordix F (2005) Propagation of displacements during a generalization process. *Bulletin du comité cartographique francais n°186 de decembre*.
- [53]. Li H, Guo Q, Liu J (2005) Rapid algorithm of building typification in web mapping. *Symposium international sur la modélisation spatio-temporelle, Raisonnement spatial, analyse, exploration de données et fusion de données, Chine*.
- [54]. Li J, Ai T, Liu P, Yang M (2017) Continuous Scale Transformations of Linear Features Using Simulated Annealing-Based Morphing, *ISPRS*. 6:242.
- [55]. LI Z, YAN H, AI T, CHEN J (2004) Automated building generalization based on urban morphology and Gestalt theory. *International Journal of Geographical Information Science*. 18 :513-534.
- [56]. Li ZL (2007) Essential operations and algorithms for geometric transformations in digital map generalization. *International Cartographic Conference, Russie*.
- [57]. Lopez FJA, Balboa JLG, Gordo JFR (2005) Segmentation of lines by means of douglas peucker applied to effective areas of the visvalingam and whyatt algorithm. (ICC) *International Cartographic Conference, Espagne*.
- [58]. Lopez FJA, Balboa JLG (2008) Generalization-oriented road line segmentation by means of an artificial neural network applied over a moving window. *Pattern Recognition*. 41(5): 1593-1609.
- [59]. Luscher P, Weibel R, Burghardt D (2009) Integrating ontological modelling and Bayesian inference for pattern classification in topographic vector data. *Computers, Environment and Urban Systems*. 33:363-374.
- [60]. Mackaness WA, Ruas A, Sarjakoski LT (2007) *Generalisation of cartographic information: cartographic modelling and applications*. Elsevier & ICA.

- [61]. McMaster RB (1987) Automated line generalization. *Cartographica*. 24(2): 74-111.
- [62]. Muller JC (1987) Fractal and Automated Line Generalization. *The Cartographic Journal*. 24(1): 27-34 DOI: 10.1179/caj.1987.24.1.27
- [63]. Muller JC (1990) The Removal of Spatial Conflicts in Line Generalization. *Cartography and Geographic Information Systems*. 17(2): 141-149 DOI: 10.1559/152304090783813817
- [64]. Mustiere S (1998) GALBE: Adaptive Generalisation. The need for an Adaptive Process for Automated Generalisation, an Example on Roads. GIS Planet, Portugal.
- [65]. Mustière S (2001) Apprentissage supervisé pour la généralisation cartographique. Thèse de doctorat. Université de Pierre et Marie Curie, Paris.
- [66]. OEEPE. 1997. Generalisation algorithms templates. Rapport Technique : Working group on generalisation.
- [67]. Perkal J (1958) An attempt at objective generalization. *Geodezja I Kartografia* 7(2): 130–142
- [68]. Plazanet C (1996) Enrichissement des bases de données géographiques : analyse de la géométrie des objets linéaires pour la généralisation cartographique (application aux routes). Thèse de doctorat. Université de Gustave Eiffel, Paris.
- [69]. Pilehforooshha P, Karimi M (2019) An integrated framework for linear pattern extraction in the building group generalization process. *Geocarto International* 34(9):1000-1021.
- [70]. Pilehforooshha P, Karimi M (2020) A local adaptive density-based algorithm for clustering polygonal buildings in urban block polygons. *Geocarto International* 35(2):1-27.
- [71]. QGIS.org. Coordinate Reference Systems. [https://docs.qgis.org/3.16/en/docs/gentle\\_gis\\_introduction/coordinate\\_reference\\_systems.html#coordinate-reference-system-crs-in-detail](https://docs.qgis.org/3.16/en/docs/gentle_gis_introduction/coordinate_reference_systems.html#coordinate-reference-system-crs-in-detail). Accédé en Mai 2021
- [72]. Qingsheng G, Hongsheng L, Jiping L (2005) Buildings Progressive Typification Based on Weighted Mesh Simplification. *Geomatics and Information Science of Wunan University*. (4):1671-8860.
- [73]. Regnauld N (2003) Algorithms for the amalgamation of topographic data. Workshop sur les progrès de la généralisation cartographique automatisée, Royaume-Uni.

- [74]. Rezigui R, Ben Sassi Y (2019) Etude et analyse de l'algorithme de Fortune pour le calcul du diagramme de Voronoi de points. Mémoire de master. Université de Laghouat, Laghouat.
- [75]. Ruas A (1999) Modèle de généralisation de données géographiques à base de contraintes et d'autonomie. Thèse de doctorat. Université de Gustave Eiffel, Paris.
- [76]. Sabo M.N. Bédard Y. Moulin B. Bernier E. (2008) Toward self-generalizing objects and on-the-fly map generalization. *Cartogr. Int. J. Geogr. Inf. Geovis.* 43, 155–173.
- [77]. Saux E (2003) B-spline Functions and Wavelets for Cartographic Line Generalization. *Cartography and Geographic Information Science.* 30(1):33-50 DOI: 10.1559/152304003100010938
- [78]. SEOS. Science Education through Earth Observation for High Schools. Système d'information géographique (SIG). <https://seos-project.eu/agriculture/agriculture-c03-s01.fr.html>. Accédé en Mai 2021.
- [79]. Shamos M. Hoey D. (1975) Closest-Point Problems. 16th Annual Symposium on Foundations of Computer Science, p. 151-162.
- [80]. Skopeliti A, Tsoulos L (1999) On the parametric description of the shape of the cartographic line. *Cartographica.* 36(3) :53–65
- [81]. Stefanakis E (2015) SELF: Semantically enriched Line simplification. *International Journal of Geographical Information Science.* 29(10): 1826-1844
- [82]. Steiniger S (2007) Enabling pattern-aware automated map generalization. Thèse de doctorat. Université de Zurich, Suisse.
- [83]. Tamilmani R, Stefanakis E (2017) Enriched geometric simplification of linear features. *GEOMATICA.* 71(1): 3-19
- [84]. Tienaah T, Stefanakis E, Coleman D (2015) Contextual Douglas- Peucker simplification. *Geomatica.* 69(3): 327-338
- [85]. Touya G (2011) Orchestration d'un processus multi-modèles de généralisation d'espaces géographiques hétérogènes. Thèse de doctorat. Université de Gustave Eiffel, Paris.
- [86]. Van Horn E (1986) Generalizing Cartographic Data Base. *AUTO-CARTO.* 7: 532-540.
- [87]. Visvalingam M, Whyatt JD (1993) Line generalisation by repeated elimination of points. *The Cartographic Journal.* 30(1): 46-51 DOI: 10.1179/000870493786962263

- [88]. Visvalingam M, Whelan JC (2016) Implications of Weighting Metrics for Line Generalization with Visvalingam's Algorithm. *The Cartographic Journal*. 53(3): 253-267 DOI: 10.1080/00087041.2016.1149906
- [89]. Wang X, Burghardt D (2019) A Mesh-Based Typification Method for Building Groups with Grid Patterns. *International Journal of Geo-Information* 8(4):168 DOI: 10.3390/ijgi8040168
- [90]. Ware JM, Wilson ID, Ware JA (2003) A Knowledge Based Genetic Algorithm Approach to Automating Cartographic Generalisation. *Applications and Innovations in Intelligent Systems X*. Springer, London. [https://doi.org/10.1007/978-1-4471-0649-4\\_3](https://doi.org/10.1007/978-1-4471-0649-4_3)
- [91]. Wei Z, Guo Q, Wang L, Yan F (2018) On the spatial distribution of buildings for map generalization. *Cartography and Geographic Information Science*. DOI: 10.1080/15230406.2018.1433068.
- [92]. Weibel R (1996) A Typology of Constraints to Line Simplification. *7th International Symposium on Spatial Data Processing, The Netherlands*. 2(sec.9a): 1- 14.
- [93]. White ER (1985) Assessment of Line-Generalization Algorithms Using Characteristic Points. *The American Cartographer*. 12(1): 17-28 DOI: 10.1559/152304085783914703
- [94]. Wikipedia Pattern. The free encyclopedia.<https://fr.wikipedia.org/wiki/Pattern>. Accédé en Mars 2021.
- [95]. Wilson ID, Ware JM, Ware JA (2003) A Genetic Algorithm approach to cartographic map generalisation. *Computers in Industry*. 52(3):291-304 [https://doi.org/10.1016/S0166-3615\(03\)00132-5](https://doi.org/10.1016/S0166-3615(03)00132-5)
- [96]. Woojin P, Kiyun Yu (2011) Hybrid line simplification for cartographic generalization. *Pattern Recognition Letters*. 32(9): 1267-1273.
- [97]. Yang W (2008) Identify Building Patterns. *Remote Sensing and Spatial Information Sciences*, Pékin.
- [98]. Zahn CT (1971) Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE TransComput*. 20:68–86.
- [99]. Zhang X, Ai T, Stoter J (2012) Characterization and Detection of Building Patterns in Cartographic Data: Two Algorithms. *Lecture Notes in Geoinformation and Cartography*. Springer, Berlin, Heidelberg.

[100]. Zhilin L (1988) An algorithm for compressing digital contour data. The Cartographic Journal. 25: 143-146.