



# **Guide méthodologique de production du squelette vectoriel (D2)**

Subvention financée par le Service Public de Wallonie, DGO3 et le Département de la  
Géomatique du Secrétariat Général.

Novembre 2018

## TABLE DES MATIERES

<b>1</b>	<b><u>OBJET DU WP2 .....</u></b>	<b><u>ERREUR ! SIGNET NON DEFINI.</u></b>
<b>2</b>	<b><u>APERÇU DE LA CHAÎNE DE TRAITEMENT .....</u></b>	<b><u>2</u></b>
<b>3</b>	<b><u>COMPARAISON ET ANALYSE DES BASES DE DONNÉES .....</u></b>	<b><u>3</u></b>
3.1	MÉTHODOLOGIE D'ANALYSE .....	3
3.2	BÂTIMENTS.....	4
3.3	ROUTES .....	9
3.4	COURS D'EAU .....	11
3.5	CHEMINS DE FER.....	12
<b>4</b>	<b><u>SÉLECTION DES DONNÉES À INTÉGRER DANS LE SQUELETTE VECTORIEL .....</u></b>	<b><u>13</u></b>
<b>5</b>	<b><u>INTÉGRATION DES DONNÉES DANS LE SQUELETTE VECTORIEL.....</u></b>	<b><u>16</u></b>
5.1	VOIRIES .....	16
5.2	BÂTIMENTS.....	23
5.3	COURS D'EAU .....	24
5.4	CHEMINS DE FER.....	24
<b>6</b>	<b><u>ANNEXES.....</u></b>	<b><u>25</u></b>

## 1 Introduction

Le WP2 a pour but de préparer les données vectorielles des différentes bases de données géographiques utiles à la cartographie du LC et du LU pour toute la Région wallonne. Parmi ces bases de données, plusieurs couches thématiques fournissent une information thématiquement fiable et d'une grande précision géométrique. Un squelette du territoire wallon est composé au départ de l'intégration de ces quelques géométries existantes. Les tâches subséquentes se baseront sur les géométries de ce squelette pour garantir une cohérence spatiale avec les couches authentiques planimétriquement précises.

Un nombre restreint de géométries validées / authentiques pour la définition de ce squelette a été analysé, notamment les géométries du PICC (bâtiments), de l'IGN (réseaux), de CENNIC (cours d'eau non-navigables) et d'ESP/OrthoWatch (plans d'eau). Ces géométries, une fois validées pour entrer dans la composition du squelette vectoriel, seront nettoyées, complétées et combinées dans une couche unique. Le choix de ces géométries pourra être redéfini au regard de l'analyse des besoins, de leur évolution au cours du projet et de l'analyse qualitative de celles-ci.

### Actions et livrables :

- Squelette vectoriel du territoire wallon (T+6).
- Guide méthodologique de production (T+6).

## 2 Aperçu de la chaîne de traitement

La figure 1 reprend les étapes successives de la production du squelette vectoriel sur base des données existantes en Wallonie.

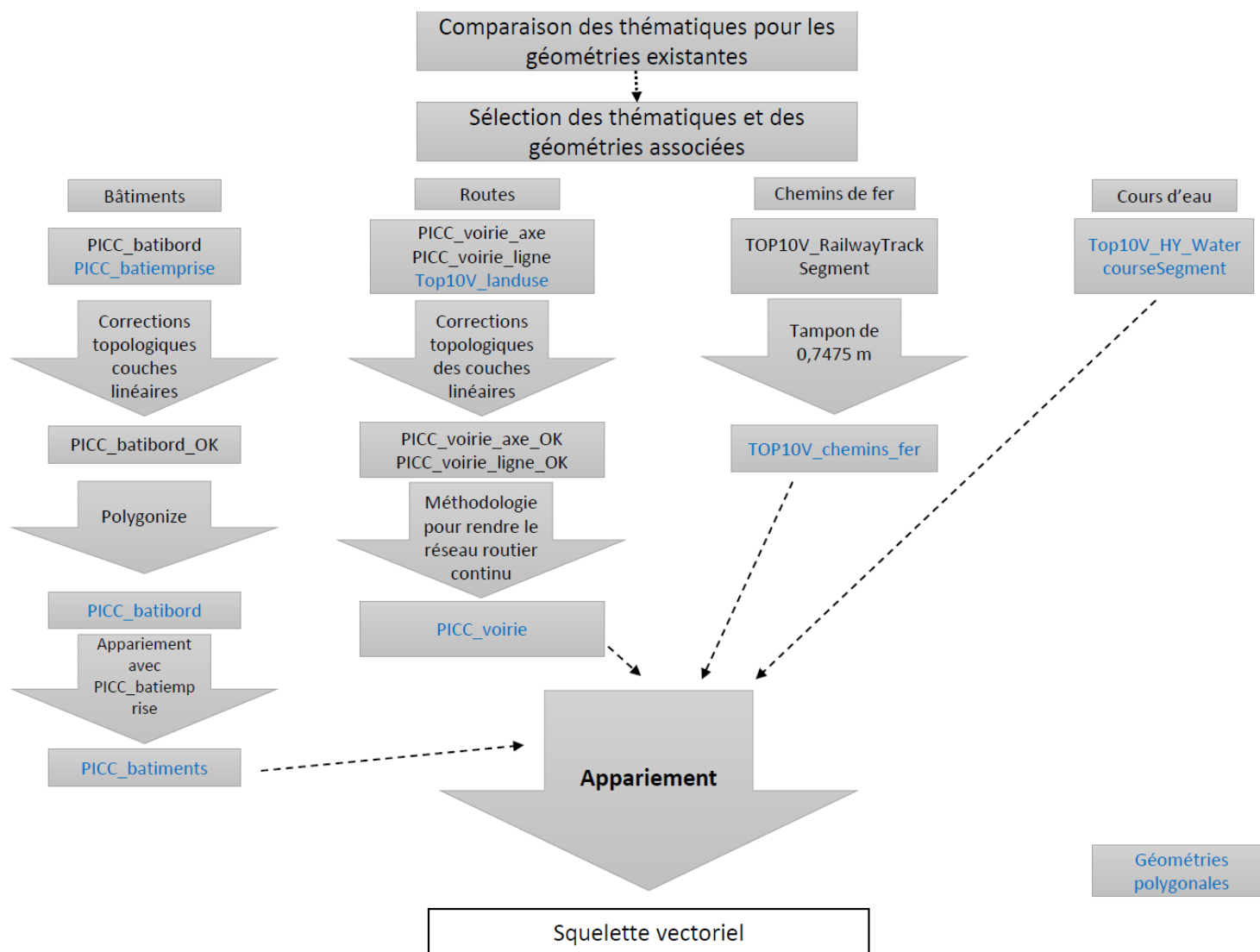


Figure 1 : chaîne de traitement du WP2 pour la création de géométries polygonales de référence

## 3 Comparaison et analyse des bases de données

Les différentes bases de données utilisées pour les analyses sont les suivantes :

- Le PICC (2017) obtenu via le SPW ;
- L'IGN/Top10V (2018) obtenu via le SPW ;
- Le cadastre (2016) ;
- L'OpenStreetMap (2015) ;
- Le CENNIC: les données utilisées pour les analyses sont les données de la dernière version qui a été produite à l'UCL. Il est possible que des modifications aient été opérées par la suite mais elles n'ont pas été prises en compte dans nos analyses ;
- Le RHW : emprise des cours d'eaux navigables (2016) et axes des cours d'eau (2012).

### 3.1 Méthodologie d'analyse

#### Zones d'analyse

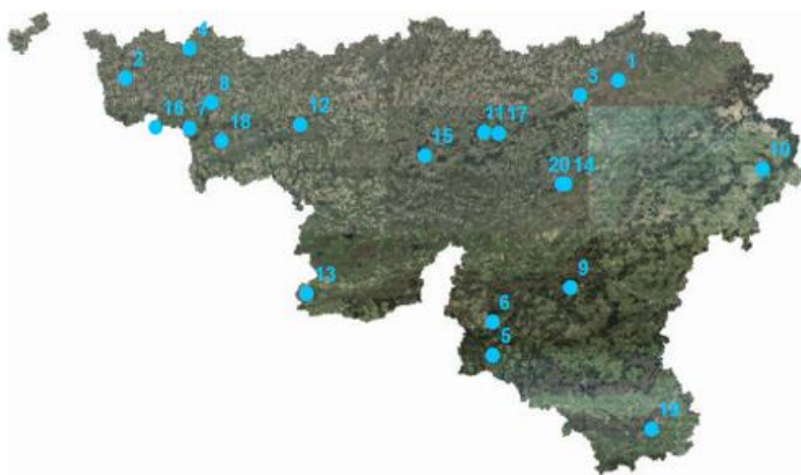
L'analyse des différentes bases de données au niveau des bâtiments et des routes. Dix zones de diamètre d'1 km ont été sélectionnées par un échantillonnage aléatoire stratifié en utilisant la densité du bâti des écotopes de LifeWatch. Le centre de chaque zone a été localisé aléatoirement (fig.2) dans chaque strate via l'outil « sélection aléatoire » de QGIS:

- 5 zones dans les écotopes densément artificialisés de LifeWatch ;
- 5 zones dans les écotopes moins densément artificialisés de LifeWatch.



Figure 2 : Localisation des zones choisies aléatoirement pour l'analyse des routes et des bâtiments. Le tableau reprend les coordonnées Lambert 1972 du centre de chaque zone.

Au niveau des cours d'eau, une sélection aléatoire de 20 éléments du RHW a été effectuée via l'outil « sélection aléatoire » de QGIS. Autour de ces 20 éléments, 20 zones d'analyses (fig.3) ont été établies suivant un diamètre d'1 km.



Zone	x	y
1	235699	146285
2	79266	147182
3	223348	141563
4	99499	156508
5	195579	59279
6	195524	69607
7	99802	131233
8	106759	139266
9	220312	80549
10	281028	118225
11	193034	129773
12	134894	132115
13	136931	78885
14	218609	113348
15	174046	122307
16	88697	131638
17	197799	129644
18	109972	127107
19	245913	35868
20	217469	113530

Système de coordonnées  
Lambert-belge 1972

Figure 3 : zones choisies aléatoirement pour l'analyse des cours d'eau. Le tableau reprend les coordonnées Lambert 1972 du centre de chaque zone.

Les thématiques suivantes ont été analysées pour les géométries existantes :

- Les bâtiments ;
- Les routes ;
- Les chemins de fer ;
- Les cours d'eau.

Les analyses effectuées, les résultats ainsi que les couches des bases de données utilisées sont présentées ci-après pour chaque thématique.

## 3.2 Bâtiments

### Données utilisées

- PICC : vTOPO.CONSTR\_BATIBORD (couche linéaire) et vTOPO.CONSTR\_BATIEMPRISE (couche polygonale) ;
- IGN : CO\_Building (couche polygonale) et CO\_ParticularPolyConstruction (couche polygonale) ;
- Cadastre (couche polygonale).

## Analyse de proximité entre les bases de données

L'analyse de proximité consiste à déterminer le pourcentage des éléments d'une base de données inclus dans les tampons des éléments homologues d'une autre base de données. Au plus le pourcentage est élevé entre deux bases de données à une distance donnée pour des tampons de faible distance, au plus elles sont proches et compatibles géométriquement. Cette analyse porte uniquement sur des bâtiments homologues afin d'évaluer uniquement la concordance géométrique et non la complétude (présence/absence de bâtiments) ou le degré de généralisation (bloc ou bâtiment isolé).

La méthodologie utilisée pour déterminer cette analyse de proximité est la suivante :

- Sélection par localisation des polygones « bâtiments » homologues des bases de données du PICC, de l'IGN et du cadastre ;
- Conversion des polygones d'une base de données en lignes ;
- Création de tampons avec une distance fixe de 25 cm, 50 cm, 1 m, 2 m et 5 m autour des polygones d'une seconde base de données ;
- Calcul du pourcentage des lignes des polygones de la première base de données incluses dans les zones tampons des polygones de la seconde base de données.

La méthodologie a été réalisée pour l'analyse de l'IGN et du PICC, ainsi que de l'IGN et du cadastre.

Les résultats sont repris dans le tableau 1.

Tableau 1 : résultats de l'analyse de proximité des couches IGN, PICC et du cadastre pour des tampons de 25 cm, 50 cm, 1 m, 2 m et 5 m

Tampons	IGN dans les tampons du PICC	PICC dans les tampons du IGN	IGN dans les tampons du cadastre
<b>Nombre de bâtiments analysés</b>	2413 polygones homologues entre le PICC et l'IGN	2413 polygones homologues entre le PICC et l'IGN	1786 polygones homologues entre l'IGN et le cadastre
<b>25 cm</b>	82,73 %	84 %	48,08 %
<b>50 cm</b>	93,16 %	94,13 %	57,65 %
<b>1 m</b>	97,75 %	98,29 %	70,47 %
<b>2 m</b>	99,02 %	99,33 %	83,3 %
<b>5 m</b>	99,07 %	99,82 %	95,84 %

## Complétude des bases de données

Le nombre de polygones présents dans chaque couche a été sommé sur les 10 zones d'étude en tenant compte de la généralisation des polygones de l'IGN (les polygones du PICC et du cadastre ont été généralisés via l'outil dissolve avec l'option « single parts »). Les résultats sont repris dans le tableau 2.

Pour la base de données PICC, les polygones de la couche PICC\_BATIBORD non homologues à ceux de la couche PICC\_BATIEMPRISE ont été additionnés aux polygones de la couche PICC\_BATIEMPRISE.

Tableau 2 : résultats de l'analyse de complétude des bases de données PICC, IGN et du cadastre au niveau des bâtiments

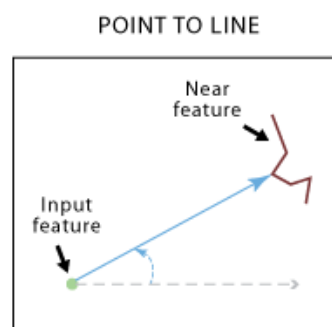
	Nombre de polygones
<b>PICC_BATIEMPRISE et PICC_BATIBORD</b>	3 403
<b>Top10V</b>	2 585
<b>Cadastre</b>	2 240

## Proximité des bases de données PICC et IGN à l'ortho 2016

Dans cette analyse, la distance des limites homologues d'un bâtiment visible sur l'ortho 2016 et celles du même bâtiment présent dans le PICC et l'IGN est calculée. Il est ainsi possible de déterminer dans quelle base de données les géométries sont les plus proches de l'ortho 2016.

La proximité est déterminée comme suit :

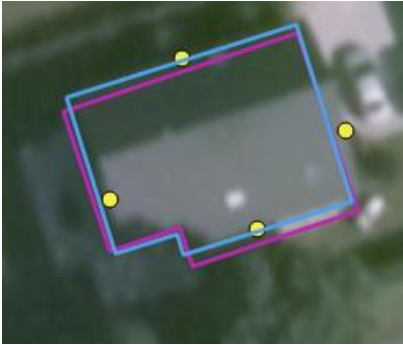


- Sélection par localisation de 15 bâtiments dans chacune des 10 zones : sur les 2413 polygones homologues entre le PICC et l'IGN, 150 ont donc été sélectionnés aléatoirement ;
- Placement de quatre points par bâtiment homologue en fonction des limites de ces bâtiments visibles sur l'ortho 2016, soit 60 points par zone.
- Vérification visuelle du placement des points pour s'assurer de ne comparer que des limites homologues au PICC et à l'IGN (exemples dans le tableau 3) ;
- Calcul de la distance entre 600 limites de polygones du PICC et de l'IGN et les points placés sur l'ortho 2016 via l'outil « Near (Analysis) »<sup>1</sup>.

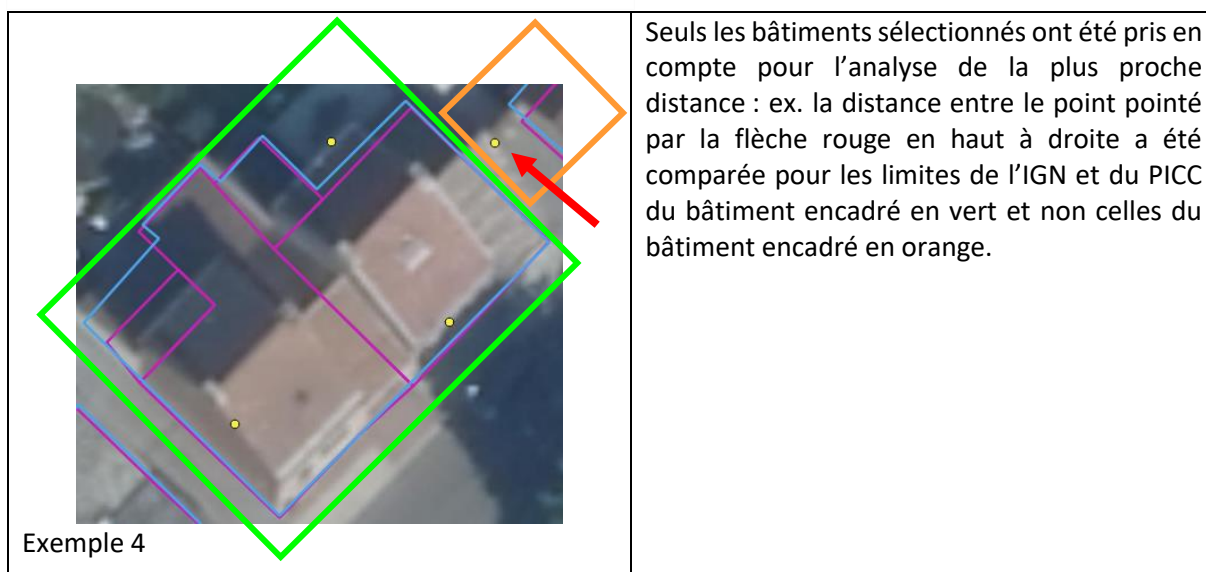


<sup>1</sup> L'outil « Near (Analysis) » calcule la distance entre un point donné et l'entité la plus proche (point, ligne ou vecteur)



Tableau 3 : exemples de placement des points aux limites des bâtiments visibles sur l'ortho 2016 homologues au PICC et à l'IGN pour l'analyse de proximité des bases de données à l'ortho 2016.

<p>Exemple 1</p> 	<p>Légende :</p> <p><b>PICC</b></p> <p><b>IGN</b></p> <p><b>Points placés sur l'ortho 2016 (jaune)</b></p>
<p>Exemple 2</p> 	<p>Les points ont été placés pour ne comparer que les lignes homologues des bâtiments homologues : les limites de l'annexe du bâtiment représenté par l'IGN (en bas à gauche de l'image) ne sont pas prises en compte pour la comparaison car le PICC ne représente pas cette annexe.</p>
<p>Exemple 3</p> 	<p>De même que pour l'exemple 2, les limites de l'annexe représentée par l'IGN en bas à gauche ne sont pas prises en compte pour la comparaison car elles ne sont pas présentes dans la représentation du bâtiment par le PICC.</p>



Calcul de la moyenne et de l'écart-type des distances les plus petites entre les limites homologues des bâtiments homologues visibles sur l'ortho 2016 et celles du PICC et de l'IGN. Les résultats sont visibles dans le tableau 4.

Tableau 4 : résultats de l'analyse de proximité des bases de données PICC et IGN à l'ortho 2016 pour la thématique des bâtiments

	Moyenne	Ecart-type
<b>PICC</b>	0,76 m	0,61 m
<b>Top10V</b>	0,78 m	0,64 m

- Analyse de la distribution des écarts (en mètres) entre les limites de bâtiments visibles sur l'ortho 2016 et celles du PICC et de l'IGN (figures 4 et 5).

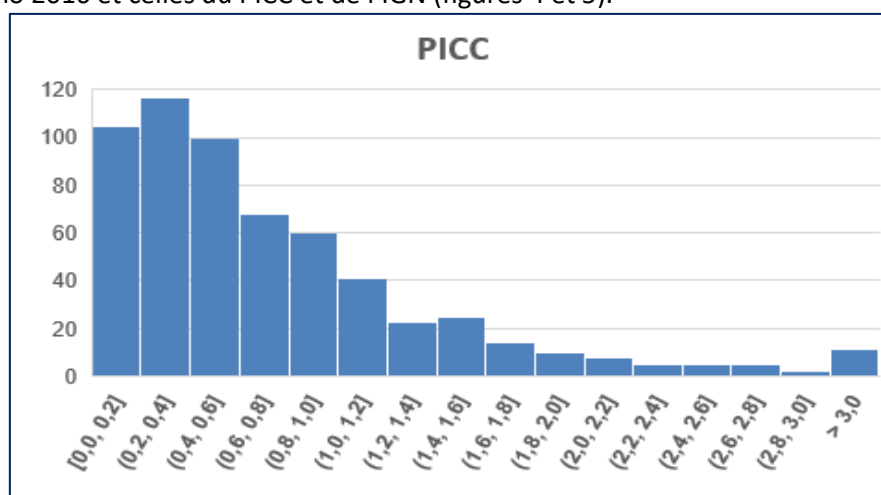


Figure 4 : distribution des écarts (en mètres) entre les limites de bâtiment visibles sur l'ortho 2016 et celles du PICC

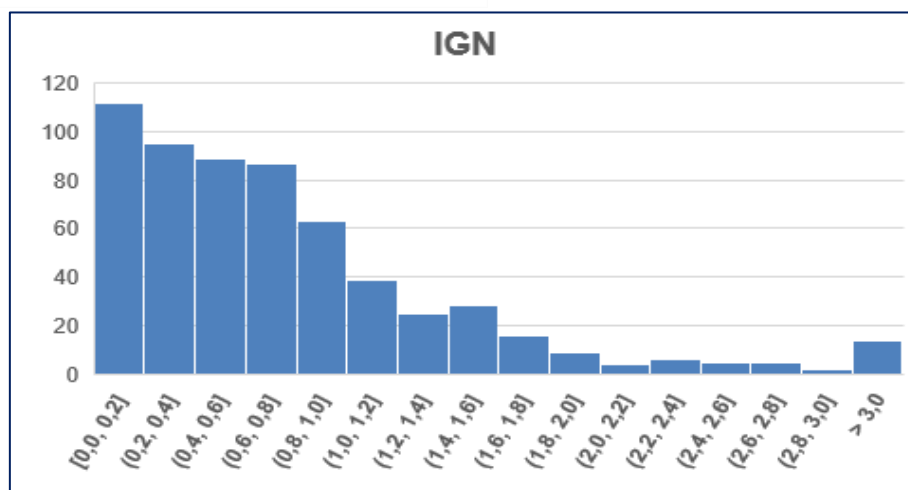


Figure 5 : distribution des écarts (en mètres) entre les limites de bâtiment visibles sur l'ortho 2016 et celles de l'IGN

### 3.3 Routes

#### Données utilisées

- PICC : vTOPO.VOIRIE\_AXE (couche linéaire de l'axe des routes) et vTOPO.VOIRIE\_LIGNE (couche linéaire de l'emprise des routes) ;
- IGN : RO\_RoadSegment (couche linéaire de l'axe des routes), RO\_DirtRoadSegment (couche linéaire de l'axe des chemins de terre) et RO\_PATH\_SEGMENT (couche linéaire des axes des chemins) et Top10V\_GIS\_LANDUSE (dernière version de la couche LANDUSE produite par l'IGN : couche polygonale de l'emprise des routes);
- OSM (2015) : couche linéaire des axes des routes.

#### Complétude des bases de données

Pour comparer la complétude des bases de données, la somme de la longueur totale des axes des routes a été calculée dans les 10 zones d'analyse. La longueur totale des routes est reprise dans le tableau 5 pour chaque base de données analysée.

Tableau 5 : résultats de l'analyse de la complétude des routes pour les bases de données IGN, cadastre et PICC



	Longueur des routes
<b>Top10V</b>	80 352,6 m
<b>Cadastre</b>	62 439,5 m
<b>PICC</b>	60 429,7 m

#### Analyse de proximité des bases de données avec l'ortho 2016

De même que pour l'analyse de proximité des bâtiments des bases de données avec l'ortho 2016, la distance entre les limites homologues des bords de routes visibles sur l'ortho 2016 et celles des bords de routes du PICC et de l'IGN est calculée (l'OSM n'est pas pris en compte car la couche utilisée est une couche représentative des axes et non des bords de route). Il est alors possible de déterminer dans quelle base de données, les géométries sont les plus proches de l'ortho 2016. Les étapes de l'analyse de proximité pour les routes sont les suivantes :

- Sélection par localisation de 30 segments de routes homologues au PICC et à l'IGN dans les 10 zones urbaines ;
- Placement de deux points aux limites des bords de routes visibles sur l'ortho 2016 pour les segments sélectionnés, soit 60 points par zone.
- Vérification visuelle des points en fonction des limites des couches PICC et IGN pour s'assurer de ne comparer que des limites homologues (exemples dans le tableau 6).
- Calcul de la distance entre 600 segments de bords de route de l'IGN et du PICC et les points placés aux limites des bords de routes visibles sur l'ortho 2016 via l'outil « Near (Analysis) ».

Tableau 6 : exemples de placement des points aux limites des routes visibles sur l'ortho 2016 homologues au PICC et à l'IGN pour l'analyse de proximité des bases de données à l'ortho 2016.

	<p>Légende :</p> <p><b>PICC</b></p> <p><b>IGN</b></p> <p><b>Points placés sur l'ortho 2016 (jaune)</b></p> <p>Exemple 1 : les points ont été placés pour des bords de routes homologues du PICC et de l'IGN.</p>
	<p>Exemple 2 : Bords de routes <b>homologues</b>. A gauche de l'image, le PICC ne dessine pas la route ; aucun point n'a donc été placé à cet endroit.</p>

- Calcul de la moyenne et de l'écart-type des distances les plus petites entre les limites homologues routes visibles sur l'ortho 2016 et celles du PICC et de l'IGN. Les résultats sont repris dans le tableau 7.

Tableau 7 : résultats de l'analyse de proximité des bases de données PICC et IGN à l'ortho 2016 pour la thématique des routes

	Moyenne	Ecart-type
PICC	0,35 m	0,35 m
Top10V (couche landuse)	0,88 m	0,63 m

## 3.4 Cours d'eau

### Données utilisées

- PICC : vTOPO.HYDROG\_AXE (couche linéaire de l'axe des cours d'eau) et vTOPO.HYDROG\_BORD (couche linéaire de l'emprise des cours d'eau) ;
- IGN : HY\_WatercourseSegment (couche linéaire de l'axe des cours d'eau) et HY\_WatercourseSurface (couche polygonale de l'emprise des cours d'eau) ;
- RHW : couche polygonale de l'emprise des cours d'eau navigables et couche linéaire des axes des cours d'eau ;
- CENNIC : dernière version de la couche linéaire des axes des cours d'eau produite à l'UCL.

### Complétude des bases de données

La somme de la longueur totale des axes des cours d'eau dans les 20 zones hydro a été calculée. Les résultats de la longueur totale pour chaque base de données sont repris dans le tableau 8.

Tableau 8 : résultats de l'analyse de la complétude des cours d'eau pour les bases de données IGN, cadastre et PICC

	Longueur des cours d'eau
Top10V	52 842,5 m
PICC	40 346 m
RHW	63 235,4 m
CENNIC	73 961,7 m

### Analyse de proximité des bases de données avec l'ortho 2016

La distance des limites homologues des bords de cours d'eau navigables visibles sur l'ortho 2016 et ceux présents dans le PICC, l'IGN et le RHW est calculée. Il est ainsi possible de déterminer dans quelle base de données, les géométries sont les plus proches de l'ortho 2016. Les étapes de l'analyse de proximité pour les cours d'eau navigables sont les suivantes :

- Sélection aléatoire de 100 segments des cours d'eau navigables du RHW sur toute la Wallonie. Une vérification visuelle par la suite a permis de sélectionner 86 segments homologues entre les bases de données PICC, IGN et RHW parmi les 100 sélectionnés de base ;
- Pour chaque segment sélectionné, placement d'un point à une limite du bord de cours d'eau visible sur l'ortho 2016.

- Vérification visuelle du placement des points pour s’assurer de ne comparer que les limites homologues aux bases de données ;
- Calcul de la distance entre 86 bords de cours d’eau de chaque base de données (PICC, IGN et RHW) et les points placés sur les bords de ces cours d’eau visibles sur l’ortho 2016 via l’outil « Near (Analysis) » ;
- Calcul de la moyenne et de l’écart-type de la distance la plus petite entre les limites homologues de bords de cours d’eau navigables visibles sur l’ortho 2016 et celles du PICC, de l’IGN et du RHW. Les résultats sont visibles dans le tableau 9.

Tableau 9 : résultats de l’analyse de proximité des bases de données PICC, IGN et RHW à l’ortho 2016 pour les cours d’eau navigables

	Moyenne	Ecart-type	Variance
<b>PICC</b>	1,18 m	1,1 m	1,2 m
<b>Top10V</b>	0,69 m	0,56 m	0,31 m
<b>RHW</b>	7,1 m	7,54 m	57 m

## 3.5 Chemins de fer

### Données utilisées

- PICC : vTOPO.FERROV\_LIGNE (couche linéaire des bords des chemins de fer) ;
- IGN : RA\_RailwayTrackSegment (couche linéaire des axes des voies de chemins de fer) ;
- OSM (2015) : couche linéaire des axes des rails de chemins de fer.

Pour l’OSM et l’IGN, un tampon de 0,7175 m (écartement des rails en Belgique = 1,435 m) est créé à gauche et à droite des axes des chemins de fer afin de représenter l’emprise des chemins de fer (= rails) et afin de pouvoir comparer la concordance planimétrique des différentes bases de données entre elles et avec l’ortho 2016.

### Proximité des bases de données de l’IGN et de l’OSM aux données du PICC

Une analyse des distances entre les limites homologues des rails du PICC et les limites de l’IGN et de l’OSM est réalisée en 3 étapes :

- Sélection de 55 segments de rails du PICC sur toute la Wallonie ;
- Vérification visuelle de la sélection pour s’assurer de ne comparer que des limites homologues au PICC, à l’IGN et à l’OSM ;
- Placement d’un point par segment de rail sélectionné du PICC ;
- Calcul des distances les plus petites entre les points sur les rails sélectionnés du PICC et les rails homologues issus des tampons de l’OSM et de l’IGN via l’outil « Near (Analysis) » et déduction de la moyenne et de l’écart-type des distances pour chaque base de données. Les résultats de l’analyse sont repris dans le tableau 10.

Tableau 10 : résultats de l'analyse de proximité des couches IGN et OSM aux données de la couche PICC pour les chemins de fer

	Moyenne	Ecart-type
Top10V	0,16 m	0,28 m
OSM	0,86 m	0,73 m

#### Analyse de proximité des bases de données avec l'ortho 2016

La distance entre les rails de chemins de fer visibles sur l'ortho 2016 et ceux présents dans le PICC, l'IGN et le OSM est calculée. Il est ainsi possible de déterminer dans quelle base de données, les géométries sont les plus proches de l'ortho 2016. Les étapes de l'analyse de proximité pour les chemins de fer sont les suivantes :

- Nouvelle sélection aléatoire de 55 segments de rails du PICC sur toute la Wallonie homologues avec les rails de l'IGN et de l'OSM ;
- Vérification visuelle de la sélection pour s'assurer de ne comparer effectivement que des segments homologues ;
- Placement d'un point sur les rails visibles sur l'ortho 2016 correspondant aux segments sélectionnés aléatoirement ;
- Calcul des distances les plus petites (Near (Analysis)) entre les points placés sur l'ortho 2016 et les rails du PICC, de l'IGN et de l'OSM.
- Calcul de la moyenne et de l'écart-type des données repris dans le tableau 11.

Tableau 11 : résultats de l'analyse de proximité des bases de données PICC, IGN et OSM à l'ortho 2016 pour les chemins de fer

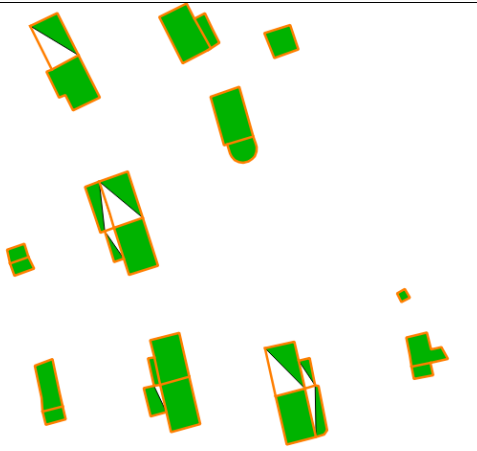
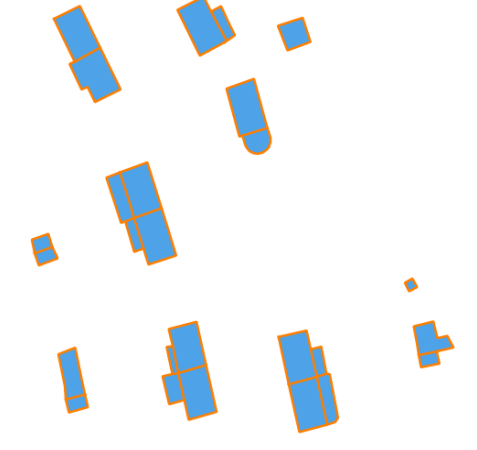
	Moyenne	Ecart-type
PICC	0,36 m	0,73 m
Top10V	0,25 m	0,42 m
OSM	0,93 m	0,96 m

## 4 Sélection des données à intégrer dans le squelette vectoriel

Au niveau des **bâtiments**, les données du PICC sont plus complètes que celles de l'IGN et du cadastre. En outre, à 1,10 m, le PICC est compatible géométriquement avec les données de l'IGN et compatible à 2 m avec l'ortho 2016 dans 95 % des cas.

Cependant, il existe des erreurs dans la topologie de la couche linéaire des bâtiments du PICC, à savoir que toutes les lignes ne sont pas fermées. D'après les quelques tests que nous avons effectués, la majorité des erreurs peuvent être corrigées en utilisant une tolérance de 25 cm pour capturer automatiquement des sommets voisins. Cela n'affecterait pas la précision géométrique dans le cadre du projet Walous, vu que cela correspond la résolution maximale des ortho 2016 :



	<p>En orange, la couche vTOPO.CONSTR_BATIBORD ; En vert, l'application de l'outil « conversion des lignes en polygones » à la couche vTOPO.CONSTR_BATIBORD ;</p> <p>On remarque que toutes les lignes ne sont pas fermées, ce qui implique que la conversion des lignes en polygones ne se fait pas correctement. Le rendu d'emprise des bâtiments n'est pas celui attendu.</p>
	<p>Via un outil de capture automatique des sommets voisins à 25 cm (v.clean de GRASS7 avec l'option « snap »), on voit, en orange, que la majorité des lignes a été fermée et, en bleu, suite à la conversion des lignes en polygones, qu'il a été possible de produire des polygones représentant l'emprise correcte des bâtiments.</p>

Il a donc été proposé d'utiliser les données du PICC pour la représentation des bâtiments dans le squelette vectoriel tout en améliorant la couche linéaire des bâtiments du PICC.

Concernant les **routes**, selon l'analyse de complétude des bases de données, il est remarqué que l'IGN et l'OSM possèdent une longueur totale de routes plus élevée que le PICC. Cela s'explique car ces deux bases de données restituent l'information des routes privées et en zones couvertes contrairement au PICC. De plus, l'IGN possède une couche représentant les chemins de terre qui ne sont pas représentés complètement dans le PICC et dans l'OSM.

Cependant, malgré une bonne complétude, l'OSM ne possède pas des données de routes assez précises pour pouvoir être utilisées dans le squelette vectoriel. La figure 6 montre un exemple de faible précision de l'OSM : les axes de routes de l'OSM traversent des bâtiments visibles sur l'ortho 2016 (flèches rouges sur la figure 6).





Figure 6 : exemple du manque de précision de la base de données OSM (en orange) pour les axes des routes.

La couche « landuse » de l'IGN, malgré une bonne complétude également, se trouve être moins précise au niveau planimétrique que le PICC et concorde moins avec l'ortho 2016.

Les données qui constitueront le squelette vectoriel doivent être le plus précises possible. Il a alors été décidé d'utiliser les données linéaires du PICC pour la représentation des routes du squelette dans la mesure où une couche surfacique peut être produite. Il est suggéré de compléter les données avec les données surfaciques de l'IGN (pour les chemins de terre et les routes privées uniquement) s'il existe un besoin de ces données pour le projet par la suite.

Au niveau des **cours d'eau navigables**, la concordance planimétrique des bords des cours d'eau de l'IGN par rapport à l'ortho 2016 est plus grande que celle du RHW et du PICC, probablement à cause des définitions. L'IGN est compatible avec l'ortho 2016 à 1,10 m dans plus de 80% des cas et à 2 m dans 95 % des cas. Les données surfaciques de l'IGN seront prises en compte pour la représentation des cours d'eau navigables dans le squelette vectoriel.

Pour les **cours non-navigables**, il n'existe pas de données de référence en Wallonie. Le CENNIC est la base de données la plus complète car c'est déjà une compilation des trois bases de données existantes. Cependant, le CENNIC restitue uniquement l'information sur l'axe des cours d'eau. Une possibilité pour l'utilisation de ces données dans la construction du squelette serait d'imposer une largeur arbitraire sur les axes ou de n'utiliser que l'axe central. Ces données restent difficiles à intégrer dans un squelette de référence et ne seront dès lors pas prises en compte pour le squelette vectoriel.

Concernant les **chemins de fer**, l'IGN possède une meilleure concordance planimétrique par rapport à l'ortho 2016 que le PICC et l'OSM. L'écart-type des distances par rapport à l'ortho 2016 est proche de 0,4 m. On peut donc dire que l'IGN est compatible avec l'ortho 2016 à 1 m dans 95% des cas. En outre, l'IGN est complet et, contrairement au PICC, ne présente que les rails existant encore actuellement (le PICC ayant encore des rails démontés dans sa base de données). Les données de chemins de fer de l'IGN seront utilisées comme éléments structurants du squelette vectoriel.

Enfin, il avait été émis l'idée d'intégrer au squelette vectoriel certaines **utilisations spécifiques du sol**. Parmi ces utilisations, il y a les éoliennes, les poteaux électriques haute tension, les cimetières, les terrains de sport ou encore les campings. Ces données existent dans les bases de données du PICC et de l'IGN (couches telles que PICC\_v.TOPO\_EQUIPE\_SURFACE, PICC\_v.TOPO\_EQUIPE\_LIGNE,

Top10V\_CO\_ParticularZone, Top10V\_CO\_ParticularPointConstruction, etc.,) mais la complétude varie d'une base à l'autre ainsi que la représentation de ces utilisations du sol (ex : les poteaux électriques sont sous forme de points dans l'IGN et sous forme linéaire dans le PICC, les terrains de sport sont délimités par leurs barrières dans le PICC, et pour leur surface de jeu dans l'IGN, ...). Il semble compliqué d'utiliser ces données peu complètes et précises comme éléments structurants du squelette vectoriel mais leur utilisation pourrait être précisée s'il existe un réel besoin dans le projet.

## 5 Intégration des données dans le squelette vectoriel

### 5.1 Voiries

Afin d'intégrer le réseau routier du PICC au squelette vectoriel, les problèmes de continuité du réseau ont d'abord dû être gérés afin de produire une couche surfacique continue. Il a été décidé de gérer ces problèmes de manière différenciée entre les bords de routes et les bords de carrefours plus complexes.

Concernant les bords de voirie, les « trous » de continuité vont être comblés, suivant certaines règles et dans la mesure du possible, par des tampons en utilisant les axes de routes continus comme support (seuls les bords de route dont l'axe est présent seront rendus continus). Cette méthodologie est développée ci-après en 12 étapes.

Par la suite, les carrefours ont été intégrés aux bords de route et rendus polygonaux mais uniquement lorsqu'ils étaient continus. Par carrefour, nous entendons les points d'intersection des lignes d'axes des voiries et les extrémités de ces axes. La gestion des carrefours et leur intégration au réseau, ainsi que la conversion du réseau linéaire en réseau surfacique a nécessité 5 étapes développées au point 5.1.2.

Pour faciliter le traitement des données, celles-ci ont été divisées en 9 zones visibles à la figure 7. Dans chacune des zones, déterminées par un choix arbitraire des communes dans chaque province, la méthodologie pour les bords de routes et les bords de carrefours a été répétée à l'identique.

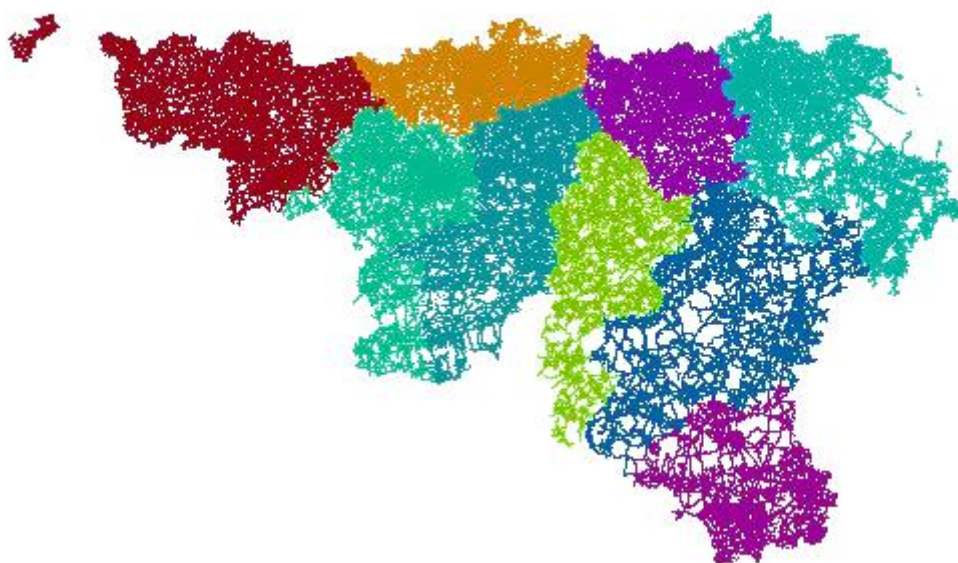


Figure 7 : zones choisies arbitrairement pour le traitement des données du réseau routier du PICC

**Etape 1 : préparation des données vectorielles**

Les couches utilisées pour intégrer les routes dans le squelette vectoriel sont les couches PICC\_VOIRIE\_LIGNE qui décrit les bords de routes (bordure, parking, passage pour piétons, trottoir, voirie-bord, etc.) et PICC\_VOIRIE\_AXE qui décrit les axes des routes communales, nationales, les axes des pistes cyclables, des rings ainsi que des chemins et sentiers.

Une sélection des lignes vectorielles a été établie pour chacune des deux couches afin de ne prendre en compte que les vecteurs utiles dans le cadre de la production de bords de routes continus : pour la couche PICC\_VOIRIE\_LIGNES, seuls les vecteurs des bords de voirie ont été sélectionnés selon l'attribut « NATUR\_CODE » = VOB, et dans la couche PICC\_VOIRIE\_AXES, les éléments « chemins et sentiers » ont été écartés.

Au niveau topologique, un nettoyage des deux couches via l'outil « fix topology error » de ArcGIS a été effectué afin d'éliminer les pseudo-nœuds et les lignes se chevauchant. Un outil similaire peut être utilisé dans QGIS, l'outil « Topology checker ». Il permet de repérer les erreurs topologiques dans une zone sélectionnée, et via l'utilisation d'un deuxième outil, v.clean de GRASS 7, il est possible de corriger ces erreurs. Les deux couches résultant du nettoyage seront appelées *axes\_propres* et *voiries\_propres* par la suite.

**Etape 2 : élimination des bords de voirie aux alentours des carrefours et aux extrémités des axes de voirie**

Pour permettre une gestion différenciée des bords de voirie et les carrefours, il a été décidé d'éliminer les bords de voirie à chaque intersection d'axes de voirie et à leurs extrémités. Ces éliminations vont également aider à faire correspondre pour chaque bord de voirie un seul axe.

Pour ce faire, un tampon de 15 m a été établi autour de chaque intersection de lignes de la couche *axes\_propres* (points d'intersection créés par l'outil « line intersections » de QGIS 3.2), ainsi qu'à l'extrémité de chaque ligne (fig.8 et fig.9). Une distance de 15 m a été choisie afin de couper un maximum de lignes aux carrefours et permettre d'attribuer à chaque bord de voirie un axe. Il faut tout de même noter qu'un tampon de 15 m ne permet pas de couper toutes les voiries aux intersections mais une grande partie. Prendre un tampon plus grand élimine trop de bords de voirie nécessaires par la suite. Les lignes de *voiries\_propres* à l'intérieur de ces tampons ont ensuite été supprimées via l'outil « différence » de QGIS. La couche résultante est appelée *voirie\_m15* par la suite.

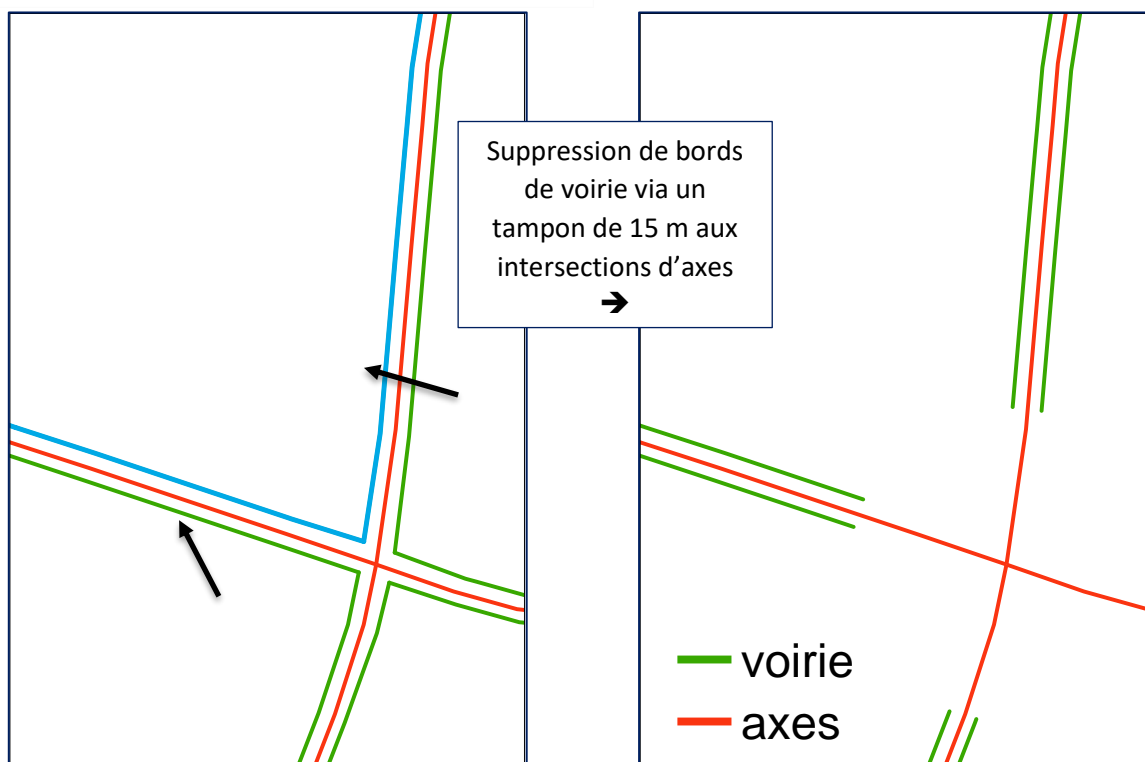


Figure 8 (g.) et 9 (d.) : en rose : axes des voiries, en vert : bord des voiries. Le bord de voirie en bleu sur l'image de gauche a deux axes de voirie lui correspondant (flèches). En éliminant les carrefours via un tampon de 15 m aux intersections, on élimine les risques de confusion des correspondances d'axes.

Une division des données des couches *axes\_propres* et *voirie\_m15* pour les 9 zones de travail déterminées arbitrairement en région wallonne a été effectuée via l'outil « select by location » de ArcGIS avec comme option de sélection « closest ». Cette option de sélection permet de sélectionner toutes les lignes les plus proches d'une zone, ce qui est particulièrement utile quand des axes de routes se situent hors des communes frontalières de la région wallonne.

#### Etape 4 : production des vertex des axes

L'outil « extract vertices » de QGIS produit les vertex de chaque axe en référençant la distance à laquelle se trouve le vertex par rapport au début de l'axe. La couche résultante est appelée *axes\_vertices*.

#### Etape 5 : calibration de l'outil Linear Referencing System<sup>2</sup> de QGIS

L'outil « Linear Referencing System » (LRS) de QGIS est une méthode de référencement spatial qui permet la localisation d'événements le long d'éléments linéaires (axes). Les événements (linéaires ou ponctuels) sont localisés comme des mesures le long des axes (en kilomètres par exemple). La première étape d'utilisation du LRS est la calibration de l'outil avec les données d'axes le long desquels seront localisés les événements ainsi que les mesures des distances de ces axes (en mètres, miles, km, etc.) sous forme d'une couche de points. La figure 10 (g.) représente l'interface de calibration du plugin. Les données visibles sur l'image sont les données entrées à titre d'exemple pour la zone 3. Une fois la calibration réalisée, il est possible de créer une couche temporaire LRS contenant les mesures des axes qui servira alors à localiser les événements/les « trous » de bords de voirie par la suite.

<sup>2</sup> Blazek et al. (2013)

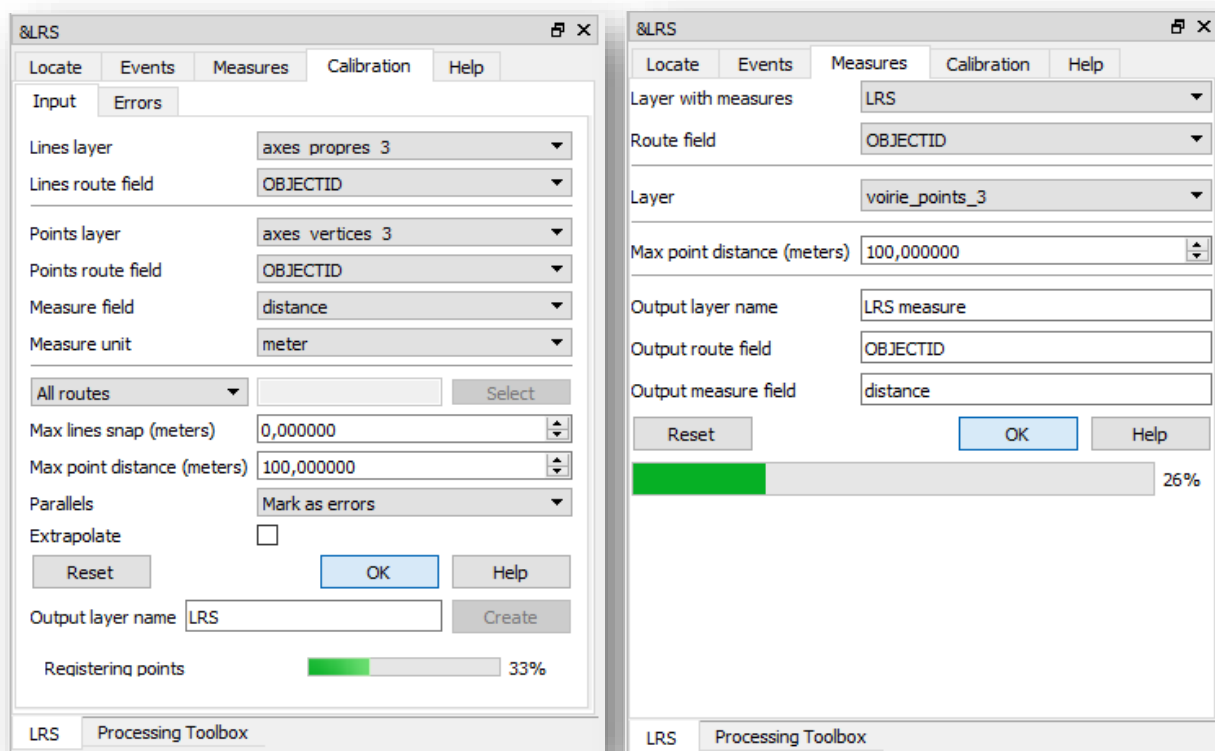


Figure 10 (g.) : calibration de l'outil LRS, (dr.) : utilisation de la fonction « mesures » de l'outil LRS

## Etape 6 : Création des points définissant les « trous » de continuité des bords de voirie et localisation de ces « trous » via le LRS

Un « trou » de continuité au bord d'une voirie va être défini par deux points : la fin d'une voirie détermine le début d'un « trou » et le début de la voirie suivante la fin de ce même « trou ». Ces points aux extrémités des voiries vont être créés via l'outil v.to.point de GRASS7 qui dessine un point aux extrémités de chaque segment de la couche *voirie\_m15*. La couche de points qui en résulte est appelée *voirie\_m15\_points*. Ces points sont ensuite localisés le long des axes grâce au plugin LRS via sa fonction « mesures ». La figure 10 (dr.) montre un exemple l'utilisation de la fonction « mesures » du plugin pour les points de la zone 3. La couche résultant est un fichier temporaire qui sera enregistré de façon permanente sous le nom *pointsLRS*.

## Etape 7 : Capture des points d'extrémités de voirie sur les axes

Comme les axes servent de support aux tampons qui combleront les bords des routes, il est nécessaire d'y reporter les points de début et fin de « trou » des voiries. Un script python (annexe 1) est utilisé. Les fonctions principales retrouvées dans ce script pour reporter les points sont les suivantes :

- QgsSpatialIndex permet de créer un index spatial sur base de la couche *axes\_propres* ;
- nearestNeighbor, basé sur l'index spatial créé, définit pour un point donné de *pointsLRS* les segments les plus proches issus de la couche *axes\_propres*. Un indice de 4 est donné afin de choisir les 4 segments les plus proches.

- `closestSegmentWithContext` permet de trouver le segment le plus proche parmi les 4 sélectionnés sous condition qu'il se situe à moins de 10 m (`mindist=100`) de ce dernier afin de ne pas reporter de points de voirie isolée (pas d'axe à proximité directe). Cette fonction donne 3 informations :
  - o la distance au carré entre ces deux points ;
  - o le point sur l'axe le plus proche du point de bord de voirie en coordonnées x-y ;
  - o le côté de l'axe où se situe le point de bord de voirie ;

Avec les informations de `closestSegmentWithContext` (coordonnées x-y du point sur le segment d'axe le plus proche), il devient possible de dessiner sur l'axe le point de bord de voirie (`.setGeometry(QgsGeometry.fromPointXY(QgsPointXY()))`) en gardant les informations utiles données dans la table d'attributs (côté de l'axe, ID de l'axe et distance entre le point de bord de voirie et l'axe le plus proche).

Ce script présente un désavantage : certains points sont reportés sur l'axe plusieurs fois. Il faut alors supprimer les géométries dupliquées via l'outil QGIS du même nom.

La couche contenant les points dessinés sur les axes est appelée *points\_snapped*.

En plus des points de voirie reportés sur les axes, des lignes sont dessinées entre ces points et les points de voirie. Elles serviront à fermer les polygones lors de la polygonisation des carrefours et des routes. Ces lignes sont stockées dans une couche appelée *lines*.

### **Etape 8 : fusion des couches ponctuelles en une unique couche de points**

Les couches *points\_snapped* et *axes\_vertices* sont fusionnées en une unique couche de points via l'outil « merge » de QGIS 3.2. Une unique couche de points est indispensable pour la création des tampons le long des axes : en effet, les points reportés des voiries délimiteront le début et la fin des tampons et les vertex permettront que les tampons suivent le tracé des axes.

### **Etape 9 : création de tampons via un script python**

Le script python de création des tampons à partir des axes se trouve à l'annexe 2. Dans ce script, pour un axe donné, tous les points situés sur cet axe sont triés suivant leur attribut « distance » acquis pour les vertex par l'outil « extract vertices » et pour les points des voiries par l'outil LRS. Une fois triés, ils sont ensuite passés en revue l'un après l'autre. Le premier vertex de l'axe est enregistré dans une liste ainsi que tous les vertex suivants jusqu'au premier point reporté de voirie se situant d'un côté de l'axe défini (gauche ou droite). Ce point est également enregistré dans la liste. La liste de points est alors convertie en polyligne (`.setGeometry(QgsGeometry.fromPolyline(polyline))`) et un tampon est créé à partir de cette polyligne (`.geometry().singleSidedBuffer()`) pour le côté choisi. La distance du tampon est déterminée par la distance du seul point de voirie clôturant la première liste. Le passage des points en revue continue jusqu'au point de voirie suivant se situant du même côté. Celui-ci est enregistré dans une nouvelle liste. Tous les vertex suivants le sont également jusqu'au point de voirie suivant qui termine la liste. Une polyligne est produite ainsi qu'un tampon dont la distance est égale à la moyenne des distances des deux points de voirie à l'axe. L'opération se poursuit ainsi de suite jusqu'au dernier vertex qui clôture la dernière polyligne et donc le dernier tampon. Ceci est effectué ensuite pour les points de voirie se situant de l'autre côté de l'axe. Une fois tous les points de l'axe passés en revue pour les deux côtés, un autre axe est passé en revue, etc.



## **Etape 10 : suppression des tampons de bords de voirie par des tampons de 12 m aux intersections et extrémités des axes de voirie et conversion de la couche produite en lignes**

Afin de ne créer que les bords de routes sans prendre en compte les carrefours, les tampons créés sont coupés via des tampons de 12 m autour des points d'intersection et d'extrémités des axes. La distance de 12 m a été choisie pour préserver une partie des tampons aux intersections afin de les fusionner avec les autres données (fig.11 : fusion des tampons coupés aux bords de voirie – voirie\_m15) en vue de produire des polygones.

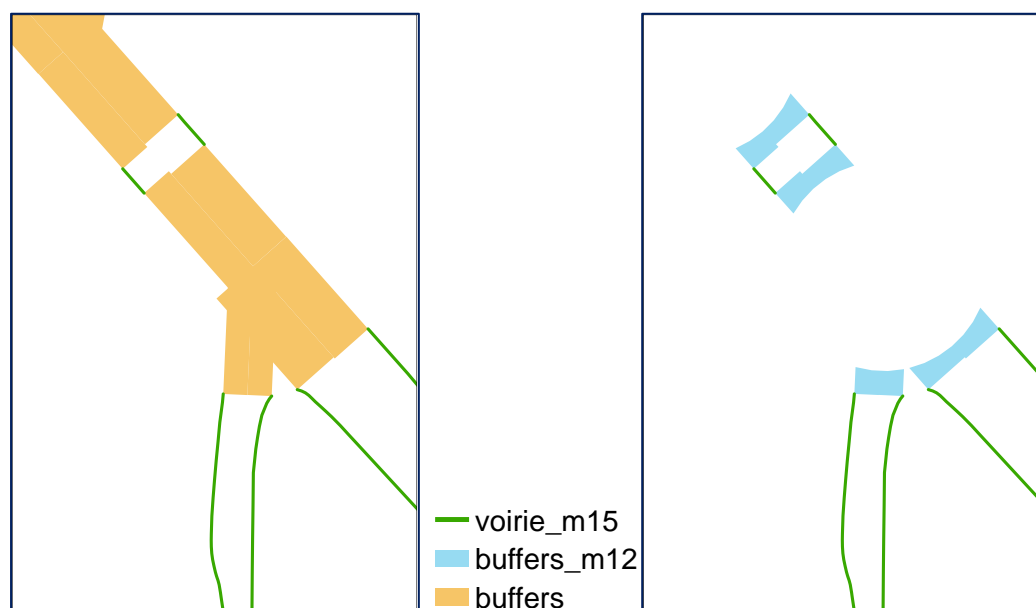


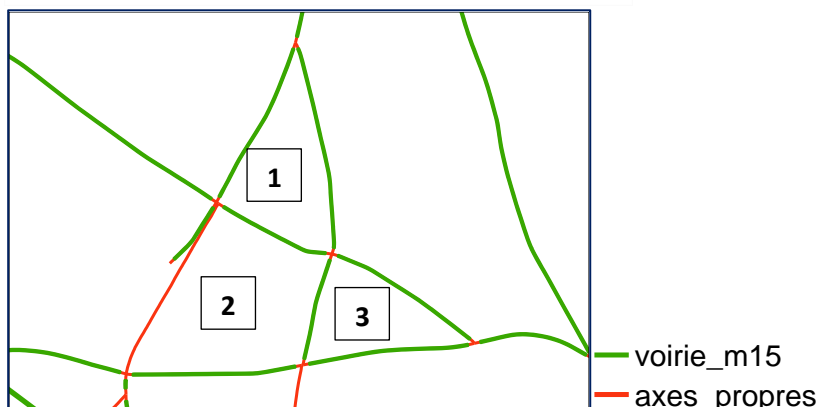
Figure 11 : suppression des tampons de bords de voirie par des tampons de 12 m aux intersections et extrémités des axes de voirie et fusion avec les bords de voirie en vert (voirie\_m15)

Les tampons de bords de route sont ensuite convertis en une couche linéaire via l'outil « polygones to lines » de QGIS. Cette action est réalisée dans le but de fusionner toutes les couches linéaires permettant par la suite la création d'une couche polygonale du réseau routier. La couche résultante est appelée *buffers\_m12\_ptol*.

## **Etape 11 : création d'une couche d'axes non continue**

Lors de la conversion du réseau routier linéaire en polygones, des polygones sont créés à l'intersection des routes sans faire partie du réseau routier (fig.12). Pour éviter cela, cette étape 11 rend discontinue la couche *axes\_propres* via l'utilisation d'un tampon de 3 m aux alentours des points d'intersection et d'extrémités des axes. Les axes discontinus sont enregistrés dans la couche *axes\_propres\_m3*.

Figure 12 : réseau routier du PICC avec les bords de voirie en vert et les axes en rose. Les chiffres 1, 2 et 3 représentent les zones où des polygones potentiels peuvent se créer suite à une conversion des lignes en polygones



## Etape 12 : fusion des couches linéaires

La fusion des couches linéaires *buffers\_m12\_ptol*, *axes\_propres\_m3*, *voirie\_m15* et *lines* est effectuée via l'outil "merge" de QGIS 3.2 (fig.13). Toutes les couches linéaires ont un intérêt dans la fermeture des lignes pour permettre ensuite la conversion des lignes en polygones. La couche résultante de la fusion est nommée *merge\_voeries* pour la suite.

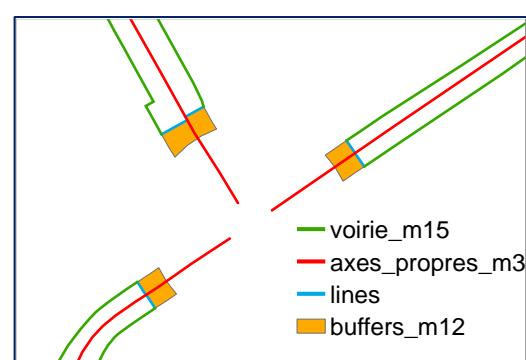


Figure 13 : fusion des couches linéaires

## Intégration des carrefours aux bords de route et polygonisation du réseau linéaire

Les carrefours sont les éléments des bords de voirie qui avaient été éliminés afin de faciliter la création des bords de routes polygonaux. Dans ce travail, les carrefours sont intégrés au réseau routier et rendus polygonaux en même temps que le réseau mais ne sont pas traités dans le but de corriger les erreurs de continuité. Il arrive donc, via la méthodologie en 5 étapes appliquée, que des erreurs soient engendrées (polygones non formés par exemple) et que, par conséquent, le réseau ne soit pas continu et surfacique sur toute la Wallonie. La forme des carrefours est en effet trop variable pour une procédure automatique basée sur des zones tampons.

### Etape 1 : Récupération des carrefours de la couche voiries\_propres

Les carrefours ont été supprimés de la couche *voiries\_propres* pour la gestion des bords de routes. Pour les récupérer, un clip a été effectué sur cette même couche via l'utilisation du tampon de 15 m réalisé autour des points d'intersection et d'extrémités des axes. La couche résultante est nommée *voirie\_carrefours* pour la suite des étapes.

### Etape 2 : Intégration des carrefours au réseau routier linéaire

Les carrefours sont intégrés au réseau routier linéaire formé au point 5.1.1 via une fusion des données linéaires *voirie\_carrefours* avec les données *merge\_voerie*. A l'issue de la fusion, il est remarqué que certaines lignes ne croisent pas d'autres lignes et ne permettent dès lors pas une fermeture des polygones. La couche résultante est alors traitée par l'outil « v.clean » de GRASS7 avec une tolérance de 25 cm qui permet qu'une ligne intercepte la ligne voisine pour peu qu'elle se situe endéans les 25 cm.



## Etape 3 : polygonisation

La couche résultante de l'étape 2 a ensuite été polygonalisée via l'outil "polygonize" de QGIS et dissoute (outil "dissolve"). La couche reprenant les polygones des carrefours est appelée *PICC\_carrefours*.

Le réseau est désormais continu dans la mesure du possible sur les bords de routes et au niveau des carrefours. Les polygones sont dissous via l'outil "dissolve" d'ArcGIS en veillant à obtenir des polygones "single parts". La couche résultante est appelée *PICC\_polygones*.

## Etape 4 : sélection des polygones de routes

Les polygones formés à l'étape 3 sont des polygones de route mais également les polygones se situant à l'intersection des routes (cfr. étape 11 au point 5.1.1). Pour éliminer le plus grand nombre de ces polygones « parasites » de manière automatique, une sélection des polygones est réalisée via l'outil « sélection par localisation » de tous les polygones « complètement à l'intérieur » d'un tampon de 20 m appliqué sur les axes routiers (*axes\_propres*). Les polygones sélectionnés sont ensuite enregistrés dans la couche *PICC\_road\_network*.

## Etape 5 : ajout des polygones manquants des carrefours et vérification des polygones produits

Pour pallier aux carrefours incomplets, les carrefours ont été également créés indépendamment. Pour ce faire, un clip a été effectué sur la couche *voiries\_propres* via l'utilisation du tampon de 15 m réalisé autour des points d'intersection et d'extrémités des axes couches. Cette couche est appelée *voirie\_carrefours* pour la suite. Les axes ont également été récupérés dans la couche *axes\_propres* sur les zones de carrefour mais avec un tampon de 16 m, le tout enregistré sous *axes\_carrefours*. Ensuite, la couche *lines* (cfr étape 8 de la gestion des bords de voirie), *voirie\_carrefours* et *axes\_carrefours* ont été fusionnées en une unique couche linéaire par l'outil "merge" de QGIS (fig.14). A l'issue de la fusion, il est remarqué que certaines lignes ne croisent pas d'autres lignes et ne permettant dès lors pas une fermeture des polygones. L'utilisation de l'outil « v.clean » de GRASS 7 avec une tolérance de 25 cm va permettre qu'une ligne intercepte la ligne voisine pour peu qu'elle se situe endéans les 25 cm.

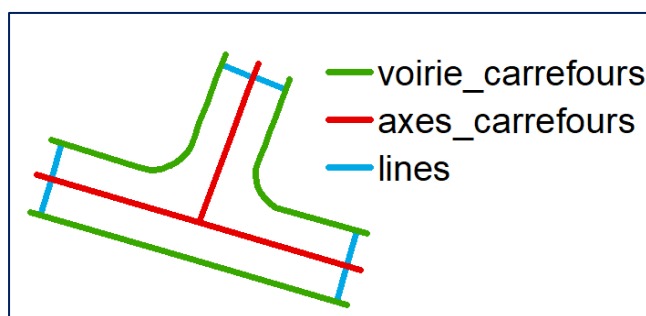


Figure 14 : fusion des couches linéaires pour les carrefours

Enfin, une vérification visuelle est effectuée pour les 9 zones en Wallonie afin de repérer d'éventuels polygones produits qui ne feraient pas partie du réseau routier wallon (ex: rond-point ou berme centrale enherbés). Les polygones des îlots directionnels, récupérés dans le *PICC\_v.TOPO\_VOIRIE\_LIGNE*, sont effacés de la couche polygonale ainsi que tous les polygones contenant des bâtiments (entièrement compris) de la couche *PICC\_CONSTR\_BATIEMPRISE*. Les erreurs restantes seront supprimées manuellement dans la mesure du possible.

## 5.2 Bâtiments

### Préparation des données vectorielles linéaires

Il existe deux couches de bâtiments du PICC : *PICC\_CONSTR\_BATIBORD*, une couche linéaire pour le bord des bâtiments et *PICC\_CONSTR\_BATIEMPRISE*, une couche polygonale de l'emprise des

bâtiments. La couche polygonale n'étant pas complète sur toute la Wallonie, il a été décidé de la compléter avec les bâtiments de la couche `PICC_CONSTR_BATIEMPRISE`. Cette couche doit être converties en polygones et pour ce faire, un nettoyage préliminaire est nécessaire pour permettre la fermeture des lignes. Ces étapes sont réalisées via l'outil « simplify » de ArcGIS et v.clean (snap de 25cm) de GRASS7 avec une tolérance de 25 cm. La couche linéaire `PICC_CONSTR_BATIEMPRISE` nettoyée est ensuite convertie en polygones.

#### 5.2.1 Création d'une couche unique polygonale

Une sélection des bâtiments semblables des deux couches du PICC est effectuée et les duplicatas (sélectionnés par localisation via l'outil « sélection par localisation » de QGIS et l'option « intersect ») sont supprimés de la couche `PICC_CONSTR_BATIEMPRISE`. Les deux couches sont ensuite appariées en une unique couche polygonale, la couche *PICC\_batiments* qui sera appariée aux autres couches thématiques pour produire le squelette vectoriel.

### 5.3 Cours d'eau

Pour les cours d'eau, il a été décidé que seuls les cours d'eau navigables de l'IGN sont pris en compte pour le squelette vectoriel. Après un découpage de la couche `HY_WatercourseSurface` sur la Wallonie, elle sera appariée aux autres couches thématiques pour produire le squelette vectoriel.

### 5.4 Chemins de fer

Pour créer une couche polygonale à partir de la couche `Top10V_RA_RailwayTrackSegment` qui présente les axes des chemins de fer, un tampon est créé autour de ces d'une distance de 0,7475 m, soit la moitié de l'écartement des voies en Belgique (1,495 m). La couche créée `TOP10V_chemins_fer` est appariée aux autres couches thématiques pour produire le squelette vectoriel.

## 6 Annexes

Annexe 1 : script python pour la capture des points d'extrémités de voirie sur les axes.

```
## QGIS 3.2 ##

pointlayer=QgsVectorLayer("Z:\\PICC_roads\\pointsLRS.shp", "", "ogr")
linelayer=QgsVectorLayer("Z:\\PICC_roads\\axes_propres.shp", "", "ogr")

epsg = linelayer.crs().postgisSrid()
uri = "Point?crs=epsg:" + str(eps) +
"&field=id:integer&field=dist:double(20,2)&field=side:integer&field=OBJECTID:real&field=distance:double(20,3)&index=yes"
snapped = QgsVectorLayer(uri,'snapped', 'memory')
prov=snapped.dataProvider()

uri2 = "LineString?crs=epsg:" + str(eps) + "&field=id:integer&index=yes"
lines = QgsVectorLayer(uri2,'lines', 'memory')
prov2 = lines.dataProvider()

testIndex = QgsSpatialIndex(linelayer)
i=0

feats=[]
feats2=[]

for p in pointlayer.getFeatures():
    i+=1
    mindist = 100.
    near_ids = testIndex.nearestNeighbor(p.geometry().asPoint(),4)
    features =
linelayer.getFeatures(QgsFeatureRequest().setFilterFids(near_ids))
    for tline in features:
        closeSegResult =
tline.geometry().closestSegmentWithContext(p.geometry().asPoint())
        if mindist > closeSegResult[0]:
            closePoint = closeSegResult[1]
            mindist = closeSegResult[0]
            side = closeSegResult[3] #1 = left, -1 = right
            OBJECTID=p["OBJECTID"]
            measure=p["distance"]
            feat = QgsFeature()

            feat.setGeometry(QgsGeometry.fromPointXY(QgsPointXY(closePoint[0],closePoint[1])))
            feat.setAttributes([i,mindist,side,OBJECTID,measure])
            feats.append(feat)
```

```
prov.addFeatures(feats)
QgsProject.instance().addMapLayer(snapped)
writer=QgsVectorFileWriter.writeAsVectorFormat(snapped,"Z:\\\\PICC_roads\\points_
_snapped.shp","CP1250",QgsCoordinateReferenceSystem("EPSG:31370"),"ESRI
Shapefile")
```

Annexe 2 : script python pour la création de tampons pour combler les bords de routes manquants.

```
import math
points_OK=QgsVectorLayer("Z:\\\\PICC_roads\\points_createbuffers.shp","", "ogr")
#load layers
axeLayer=QgsVectorLayer("Z:\\\\PICC_roads\\axes_propres.shp","", "ogr")

epsg = QgsCoordinateReferenceSystem("EPSG:31370")
uri_2= "Polygon?crs=epsg:" + str(epsg) +
"&field=id:integer&field=distance:double(20,2)&field=side:integer&index=yes"
buffer= QgsVectorLayer(uri_2,'buffer', 'memory')

buf = buffer.dataProvider()
feats_2=[]

feats=[]

for a in axeLayer.getFeatures():
    expr=QgsExpression("\"OBJECTID\" = {obj}".format(obj=a["OBJECTID"]))
    linein = [point for point in
points_OK.getFeatures(QgsFeatureRequest(expr))]
    for c in linein:
        def get_measure(c):
            return c["distance"]
        points=sorted(linein, key=get_measure, reverse=False)
        distRight=0
        distLeft=0
        countRight=0
        countLeft=0
        i=0
        startLeft = 1
        startRight = 1
        newlineLeft=[]
        newlineRight=[]

        for p in points:
            i+=1
            if startLeft==1:
                newlineLeft.append(QgsPoint(p.geometry().asPoint()))
            if i==len(points):
```

```
        if countLeft >0:
            distbuffer=distLeft/countLeft
            feat.setGeometry(QgsGeometry.fromPolyline(newlineLeft))

bufferevent=feat.geometry().singleSidedBuffer(distbuffer,5,0,1,2.0)
        polygon=bufferevent.asPolygon()
        featpol=QgsFeature()
        featpol.setGeometry(QgsGeometry.fromPolygonXY(polygon))
        feats_2.append(featpol)
        newlineLeft=[]
        countLeft=0
        distLeft=0
    if startRight==1:
        newlineRight.append(QgsPoint(p.geometry().asPoint()))
        if i==len(points):
            if countRight >0:
                distbuffer=distRight/countRight
                feat.setGeometry(QgsGeometry.fromPolyline(newlineRight))
bufferevent=feat.geometry().singleSidedBuffer(distbuffer,5,1,1,2.0)
        polygon=bufferevent.asPolygon()
        featpol=QgsFeature()
        featpol.setGeometry(QgsGeometry.fromPolygonXY(polygon))
        feats_2.append(featpol)
        newlineRight=[]
        countRight=0
        distRight=0

    if p["side"]== -1:
        startLeft*=-1
        if startLeft== -1:
            countLeft+=1
            distLeft+=math.sqrt(p["dist"])
            distbuffer=distLeft/countLeft
            feat = QgsFeature()
            feat.setGeometry(QgsGeometry.fromPolyline(newlineLeft))
bufferevent=feat.geometry().singleSidedBuffer(distbuffer,5,0,2,2.0)
        polygon=bufferevent.asPolygon()
        featpol=QgsFeature()
        featpol.setGeometry(QgsGeometry.fromPolygonXY(polygon))
        feats_2.append(featpol)
        newlineLeft=[]
        countLeft=0
        distLeft=0
        if startLeft==1:
            newlineLeft.append(QgsPoint(p.geometry().asPoint()))
            countLeft+=1
            distLeft+=math.sqrt(p["dist"])
    elif p["side"]==1:
```

```
startRight*=-1
if startRight==-1:
    countRight+=1
    distRight+=math.sqrt(p["dist"])
    distbuffer=distRight/countRight
    feat=QgsFeature()
    feat.setGeometry(QgsGeometry.fromPolyline(newlineRight))
bufferevent=feat.geometry().singleSidedBuffer(distbuffer,5,1,2,2.0
    polygon=bufferevent.asPolygon()
    featpol=QgsFeature()
    featpol.setGeometry(QgsGeometry.fromPolygonXY(polygon))
    feats_2.append(featpol)
    newlineRight=[]
    countRight=0
    distRight=0
if startRight==1:
    newlineRight.append(QgsPoint(p.geometry().asPoint()))
    countRight+=1
    distRight+=math.sqrt(p["dist"])

buf.addFeatures(feats_2)
QgsProject.instance().addMapLayer(buffer)
writer=QgsVectorFileWriter.writeAsVectorFormat(buffer,"Z:\\PICC_roads\\buffer.
shp","CP1250",QgsCoordinateReferenceSystem("EPSG:31370"),"ESRI Shapefile")
```