


# WalousMàJ

## Vectorisation

|          |            |
|----------|------------|
| Version  | 1          |
| Révision | 2          |
| Projet   | WalousMàJ  |
| Date     | 28/09/2022 |

|                            | Nom             | Société      | Signature   |
|----------------------------|-----------------|--------------|---|
| Auteur                     | Damien Draime   | Aerospacelab |  |
| Relecteur /<br>Approbateur | Jonathan Denies | Aerospacelab |  |
| Approbateur                | Benoît Deper    | Aerospacelab | <i>Benoît Deper</i>   |

# Table des matières

|                                  |          |
|----------------------------------|----------|
| <b>Table des matières</b>        | <b>2</b> |
| <b>Abréviations et acronymes</b> | <b>3</b> |
| <b>1. Objet du rapport</b>       | <b>4</b> |
| <b>2. Comparaison</b>            | <b>4</b> |
| 2.1. GDAL                        | 4        |
| 2.2. GRASS                       | 5        |
| 2.3. ArcGIS                      | 5        |
| <b>3. GDAL/QGIS</b>              | <b>7</b> |
| <b>4. GRASS</b>                  | <b>7</b> |
| <b>5. ArcGIS</b>                 | <b>8</b> |

# Abréviations et acronymes

**WAL\_OCS\_2018**

Carte d'occupation du sol de la Wallonie pour 2018 générée par le projet WALOUS

**WAL\_OCS\_IA\_2019/20**

Carte d'occupation du sol de la Wallonie pour 2019 (ou 2020) générée pour ce marché

# 1. Objet du rapport

Ce rapport a pour but de détailler les différentes options pour vectoriser la carte d'occupation du sol du territoire wallon.

La taille du raster met à l'épreuve les différents algorithmes de vectorisation et peut représenter un défi majeur pour certains.

Trois algorithmes/implémentations seront comparées : deux approches open-source et une sous licence payante. La première section de ce rapport comparera les 3 approches. Ensuite, une section sera dédiée à chaque approche pour en décrire l'utilisation.

## 2. Comparaison

Plusieurs implémentations ont été considérées pour la vectorisation du raster WAL\_OCS\_IA\_2019. Elles se différencient entre elles par les options de configuration qu'elles offrent mais également par leur algorithme.

Parmi les approches considérées, deux sont en libre accès alors que la dernière n'est accessible qu'avec une licence payante. Cette dernière approche, bien que contraire à la philosophie open-source du marché WALOUS, a été retenue car elle propose une vectorisation rapide même sur des rasters de grande taille.

### 2.1. GDAL

GDAL, <https://gdal.org>, est une librairie réputée et en libre accès. Elle propose une multitude d'utilitaires pour manipuler des données raster et vectoriel.

Cette implémentation, fidèle à son principe d'open source, supporte plusieurs drivers pour la conversion en format vectoriel. Le driver open source Geopackage (GPKG) est notamment supporté. Ainsi que ceux d'ESRI et plus spécifiquement le driver FileGeoDatabase (gdb). Cependant ceux-ci requièrent de l'utilisateur une licence ESRI. La liste des drivers supportés est fournie au lien suivant : <https://gdal.org/drivers/vector/>.

Outre le vaste choix de drivers, l'implémentation a l'avantage de proposer plusieurs options de configuration pour choisir quels pixels doivent être considérés comme voisins (option : connectedness). Soit uniquement les pixels se touchant horizontalement et verticalement peuvent être considérés comme voisins, soit également ceux se touchant en diagonale.

Cependant, AEROSPACELAB a observé durant le marché de mise à jour de la carte d'occupation du sol que, bien que cette approche fonctionne parfaitement pour les zones de +/- 16km<sup>2</sup>, lorsqu'elle est appliquée sur un raster couvrant le territoire wallon avec une résolution spatiale de 1m/pixel, cette approche est très lente. Sous différentes configurations, plusieurs tentatives ont été lancées durant le marché pour vectoriser la carte d'occupation du sol mais sans jamais aboutir. En effet, même en laissant plusieurs semaines pour l'exécution, la progression n'a jamais dépassé les 50%. De plus, bien que la parallélisation soit supportée sur certains utilitaires de GDAL, elle ne l'est pas pour celle-ci. Plusieurs articles/forums exposent la lenteur de l'approche sur internet :

- <https://gis.stackexchange.com/questions/73732/why-is-gdalpolygonize-so-much-slower-than-arcgis-raster-to-polygon>
- <https://lists.osgeo.org/pipermail/gdal-dev/2015-January/040780.html>

Une solution consiste à découper le raster d'origine en tuiles et de vectoriser indépendamment chacune d'entre elles (en parallèle ou non) et ensuite de consolider tous les résultats dans une seule base de données. Le problème de cette approche est que les polygones s'étalant sur plusieurs tuiles ne seront pas unis dans la même géométrie.

Il est à noter également que c'est cette approche qui est implémentée nativement dans QGIS.

## 2.2. GRASS

La documentation de WALOUS 2018 fait référence à la librairie open source GRASS, <https://grass.osgeo.org/>. Moins populaire que GDAL, celle-ci propose néanmoins beaucoup de fonctionnalités. De plus, GRASS offre également une interface graphique facilitant sa prise en main.

L'avantage principale de cette implémentation est qu'elle propose nativement une option de simplification des polygones via la méthode des coins cassés (option : -s).

Cependant, les expérimentations avec cette approche n'ont pas été fructueuses puisque la vectorisation était également très lente pour la carte d'occupation du sol du territoire wallon. Cette approche ne représente donc pas une solution pour la vectorisation de la carte d'occupation du sol.

## 2.3. ArcGIS

Cette approche repose sur le logiciel sous licence payante ARCGIS PRO de ESRI et plus spécifiquement leur outil de geoprocessing « Raster to polygon ».

Cette approche a l'avantage de proposer plusieurs options de configuration dont une simplification des polygones et l'imposition d'une limite au nombre de vertices par polygone. Ceci permet de se rendre compatible avec la base de données ORACLE du SPW. Cependant, l'implémentation ne permet pas, à l'instar de l'implémentation de GDAL, de considéré comme pixels voisins ceux qui se touchent en diagonale. Ainsi, un grand nombre de polygones ne désignant qu'un seul pixel peuvent venir inutilement agrandir la base de données.

Selon les expérimentations faites durant le marché, cette approche est la plus rapide des trois et de très loin.

Le tableau récapitulatif suivant met en lumière les avantages et inconvénients de chacune des trois approches considérées pour ce marché :

| Approche          | Open-source | Temps d'exécution <sup>1</sup>  | Simplification des polygones supportée nativement | Limitation du nombre de vertices | Connectedness | Autres options  |
|-------------------|-------------|---|---|----------------------------------|---------------|---|
| <b>GDAL/QGIS</b>  | Oui         | - Entier : Indéterminé (> 3 semaines)<br>- Découpe par tuile : 2 heures | Non   | Non                              | Oui           | - Plusieurs drivers supportés   |
| <b>GRASS</b>      | Oui         | Indéterminé (> 2 semaines)  | Oui   | Non                              | Non           | N/A   |
| <b>ArcGIS Pro</b> | Non         | 1 heure   | Oui   | Oui                              | Non           | - Uniquement le driver FILEGEODATABASE<br>- Délimitation<br>- Alignement sur grille |

<sup>1</sup> Le temps d'exécution renseigné est pour WAL\_OCS\_IA\_2019 sur une machine avec un CPU et 16GB de RAM.

## 3. GDAL/QGIS

Bien qu'implémentée avec le langage de programmation C, la librairie offre une série de scripts en Python agissant comme interfaces à l'implémentation en C. Ainsi un script Python « gdal\_polygonize.py » ([https://gdal.org/programs/gdal\\_polygonize.html](https://gdal.org/programs/gdal_polygonize.html)) implémenté dans GDAL permet de vectoriser un raster.

Pour exécuter ce script, Python doit donc être installée ainsi que GDAL. Ensuite la commande à exécuter pour convertir le raster en GPKG est la suivante :

```
>>> gdal_polygonize.py -8 {INPUT_RASTER_NAME} {OUTPUT_VECTOR_NAME}
```

Le degré de connectedness est de 8 dans cet exemple. Ceci signifie donc que les pixels se touchant verticalement, horizontalement et diagonalement seront considérés comme voisins.

Via QGIS, cette implémentation de GDAL peut être retrouvée via l'onglet Raster > Conversion > Polygonize :

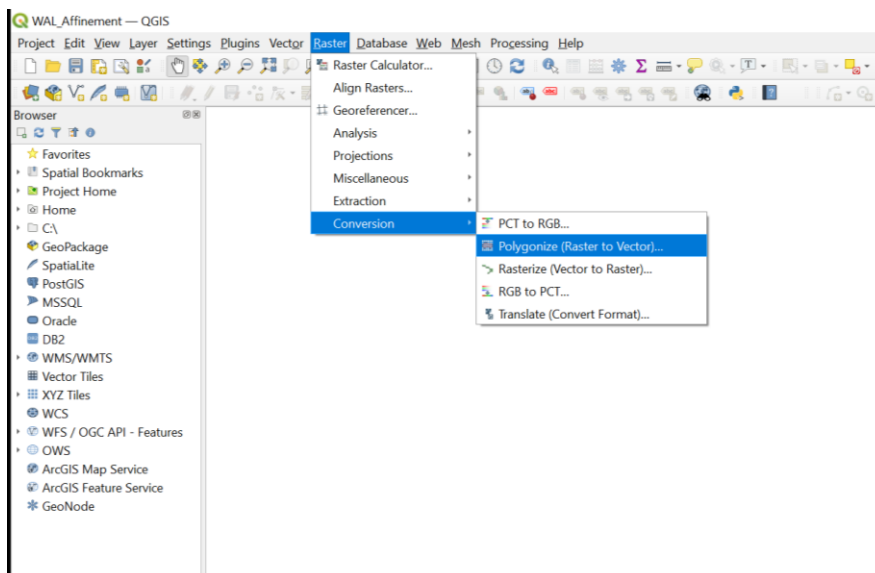


Figure 1 - Polygonize dans le menu de QGIS

## 4. GRASS

Cette approche de GRASS repose sur la fonction « r.to.vect » (<https://grass.osgeo.org/grass76/manuals/r.to.vect.html>).

L'outil GRASS doit être installée, et une fois le raster importé, la commande suivante peut être utilisée :

```
>>> r.to.vect -s input= {INPUT_RASTER_NAME} output=
{OUTPUT_VECTOR_NAME} type=area
```

L'option -s permet de simplifier les polygones via la méthode des coins cassés.

L'utilisation de l'interface graphique de GRASS offre une expérience plus intuitive. L'outil de vectorisation peut être trouvé via l'onglet Raster > Map type conversions > Raster to vector :

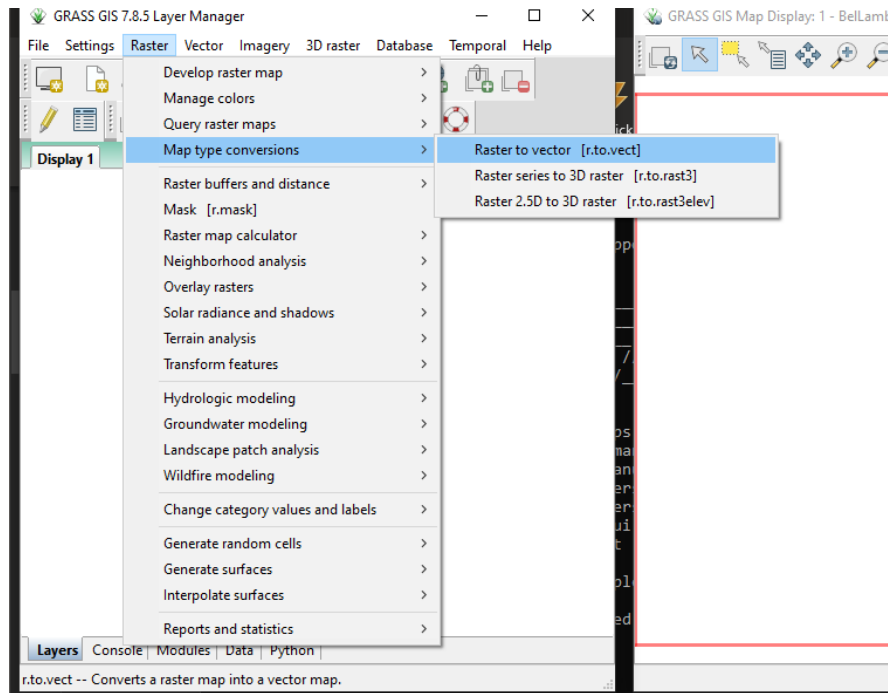


Figure 2 - Raster to vector dans le menu de GRASS

## 5.ArcGIS

Dans ArcGIS, l'outil de vectorisation (Raster to Polygon) peut être trouvé via une recherche dans le volet regroupant les outils de Geoprocessing :

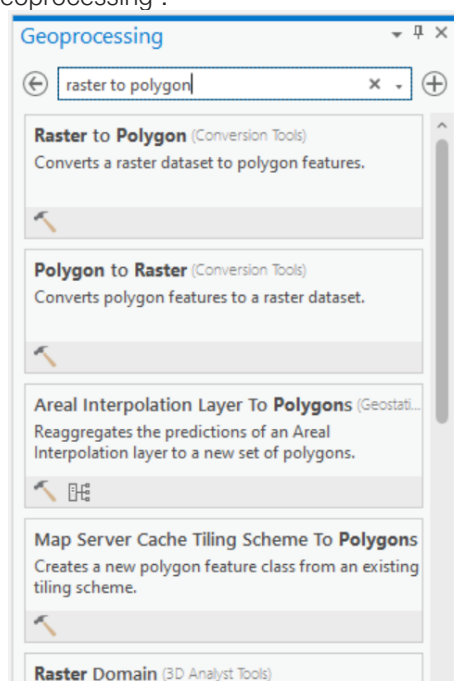


Figure 3 - Raster to Polygon dans le volet Geoprocessing

Ensuite la vectorisation peut être adaptée grâce aux options de configuration et d'environnement. Dans le cadre du marché de mise à jour de la carte d'occupation du sol, les options suivantes ont été prises :

- Pas de simplification
- Pas de « multipart features »



- Limitation à 450.000 vertices par polygone<sup>2</sup>

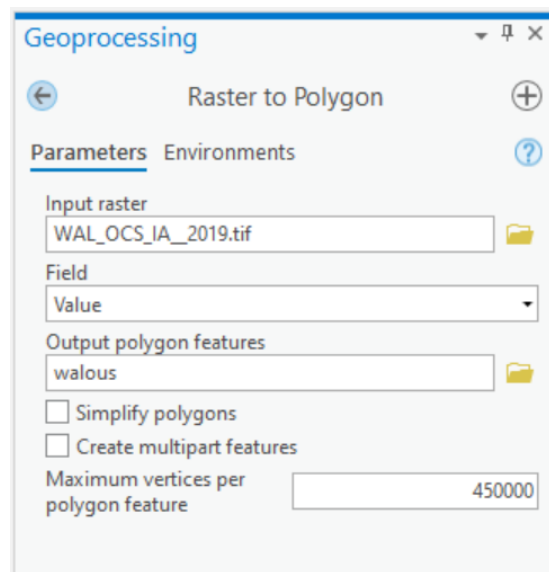


Figure 4 - Paramètres d'ArcGIS

Les autres paramètres de l'environnement ont été laissés par défaut.

<sup>2</sup> Limite imposée par la capacité par polygone 2D de la base de données du SPW