



## GéoPeuple

<i>Rapport numéro</i>	L3.2-0
<i>Titre</i>	<b>Extraction automatique des vignettes des cartes de Cassini à partir des coordonnées ponctuelles saisies lors de la vectorisation</b>
<i>Rédigé par</i>	Benoit Costes (IGN / COGIT)
<i>État (en cours / final)</i>	Final
<i>Relu par</i>	Eric Grosso, Christine Plumejeaud (IGN / COGIT)
<i>Date</i>	Juillet 2011

## 0.1 Problématique

Vu le nombre très important d'objets ponctuels saisis lors de la phase de vectorisation des cartes de Cassini, une extraction manuelle des vignettes ne semble pas raisonnable. C'est pourquoi il a été décidé de développer un processus permettant une extraction automatique des imagerie et minimisant les interventions manuelles.

On définit pour chaque type d'objet ponctuel de chaque classe un "polygone englobant optimal" correspondant au polygone dont les dimensions ont été établies d'après une étude statistique faite sur les rectangles englobants d'une sous-famille des objets de ce type.

Le processus mis en place s'appuie sur ces données statistiques et :

- Crée pour chaque objet ponctuel la géométrie polygonale définie à partir du point de saisi de l'objet et dont les dimensions correspondent au polygone englobant optimal pour ce type d'entité.
- Sauvegarde l'ensemble des polygones dans un fichier au format ESRI Shapefile.
- Crée les vignettes correspondant à un découpage du fond de carte de Cassini selon les coordonnées de chaque polygone englobant en s'appuyant sur une bibliothèque Java de manipulation d'images.

Les vignettes sont ensuite transmises au LIP6 avec leur coordonnées dans un format établi collectivement.

Le présent rapport détaille l'ensemble de ces étapes ainsi que le format d'échange utilisé. On rappelle qu'on ne s'intéresse ici qu'aux géométries ponctuelles, représentant le centre de saisie des objets défini par le dossier des spécifications (e.g pour une église, le point saisi correspond au centre du cercle de la base du bâtiment).

## 0.2 Données en entrée

### 0.2.1 Données vectorielles

Les données vectorielles saisies par l'EHESS sont enregistrées sous forme de fichiers au format ESRI shapefiles (un par classe d'objet).

### 0.2.2 Fichier de paramètres

On convient de définir le polygone englobant d'un objet ponctuel, i.e. le rectangle d'aire minimale contenant le symbole associé, par les distances caractéristiques ( $x_1$ ,  $x_2$ ,  $y_1$ ,  $y_2$ ) du centre de saisi aux quatre arêtes du polygone, comme illustré sur la figure 1. Par convention, le polygone englobant contient la base dessinée de l'objet.

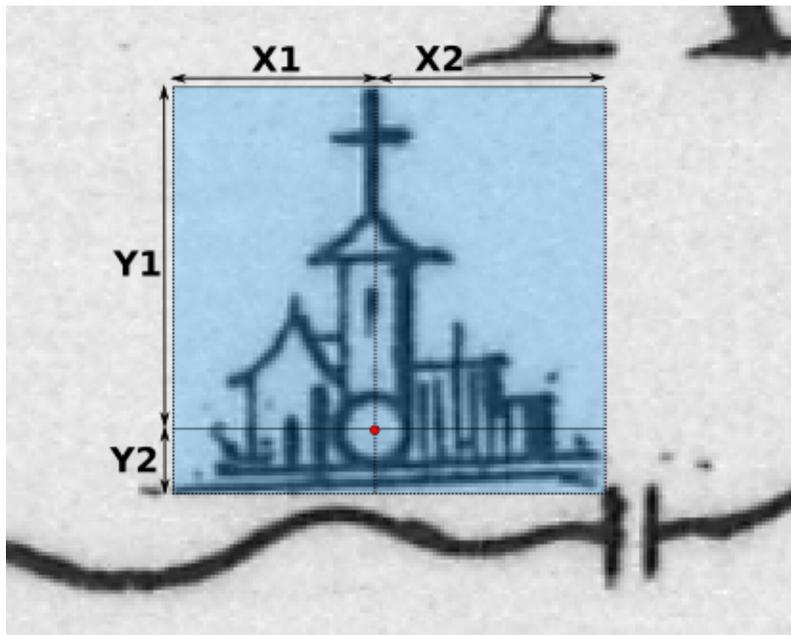


FIGURE 1 – Convention de définition du polygone englobant ici défini pour une église.

Afin d'automatiser la construction des polygones englobants, on cherche à disposer pour chaque type d'objet de statistiques (moyennes et écarts types) relatives aux dimensions des rectangles englobant les symboles associés, afin de définir pour chaque objet le polygone englobant optimal.

	A	B	C	D	E	F	G	H
1	Eglise				Eau			
2	x1	x2	y1	y2	x1	x2	y1	y2
3	60	42	74	18	18	23	19	21
4	45	59	96	14	20	20	20	19
5	46	44	90	13	14	19	17	18
6	40	41	90	11	18	19	20	19
7	45	40	91	13	18	16	18	20
8	41	48	86	14	14	17	17	18
9	50	53	100	19	20	17	19	24
10	50	49	89	13	19	18	17	15
11	48	45	92	14	18	16	17	19
12	48	60	97	15	19	19	23	22
13	56	45	90	18	17	17	22	14
14	55	44	94	13	16	16	16	15
15	61	41	84	14	19	17	18	17
16	56	63	84	16	17	16	22	16
17	45	58	95	13	20	19	19	19
18	48	45	83	22	17	16	17	18
19	63	47	83	19	19	17	22	20
20	63	51	87	10	15	16	17	16
21	57	41	87	15	19	16	20	18
22	42	49	97	16	20	17	19	17
23	52	43	92	19	18	19	20	18
24	59	66	92	20	16	20	19	18
25	47	57	86	17	18	17	19	16
26	49	62	85	19	19	20	20	18
27	47	51	84	17	17	19	22	21
28								
29								
30								
31	NB ECH							
32	25				25			
33								
34								
35	SIG x1	SIG x2	SIG y1	SIG y2	SIG x1	SIG x2	SIG y1	SIG y2
36	6,87943796	7,88077830	5,79741322	3,02379453	1,75594229	1,79536440	1,95106808	2,31444738
37								
38								
39	moy x1	moy x2	moy y1	moy y2				
40	50,92	49,76	89,12	15,68	17,8	17,84	19,16	18,24
41	max x1	max x2	max y1	max y2				
42	63	66	100	22	20	23	23	24
43	min x1	min x2	min y1	min y2				
44	40	40	74	10	14	16	16	14

FIGURE 2 – Statistiques sur les dimensions des polygones englobants pour les églises et les moulins à eau.

Ces données sont ensuite classées dans un fichier au format XML et enregistrées par type. Une première étude montre que les tailles des symboles pour un même type donné varient considérablement d'une feuille de Cassini à l'autre. Les symboles sont par exemple plus petits sur la feuille d'Agen que sur celle de Reims en raison de leur plus grande densité : pour les églises, on trouve des rectangles dont les dimensions approximatives sont respectivement de 120\*100 pixels pour la région de Reims et de 80\*85 pixels pour Agen.

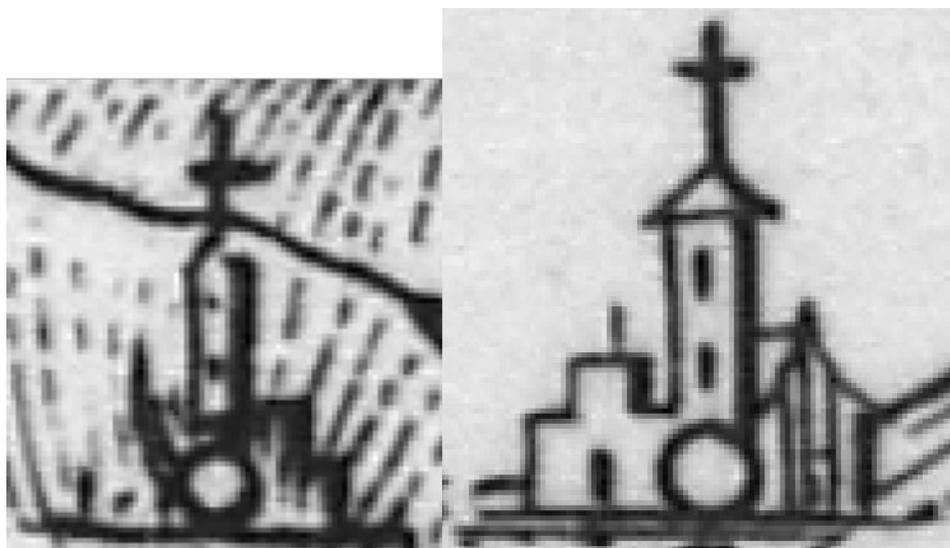


FIGURE 3 – Différentes tailles de symboles (à gauche Agen, à droite Reims).

Il est donc difficile de définir des rectangles de taille optimale en effectuant des statistiques sur les 4 zones simultanément. C'est pourquoi il a été créé un fichier de paramètres par zone. Ces fichiers sont ensuite lus par le programme afin d'extraire automatiquement les vignettes du fond de carte. Un réajustement manuel sous Qgis des dimensions des polygones peut s'avérer nécessaire au cas par cas (symbole trop petit, trop grand, mal dessiné, etc.). Une opération de recadrage (« rectangularisation ») est appliquée par la suite par notre programme afin d'obtenir en sortie des vignettes de géométrie rectangulaire (c.f figure 5).

## 0.3 Méthode

### 0.3.1 Création des polygones englobants

Le programme, écrit en Java, scanne tout un répertoire et en liste tous les shapefiles. Une boucle itérative se charge alors d'ouvrir chaque fichier vectoriel (chaque classe) afin d'en charger tous les objets. Un objet de géométrie polygonale est alors créé et les coordonnées de ses sommets sont calculées à partir de la géométrie de l'objet ponctuel (centre de saisie) et des dimensions du rectangle englobant moyen définies dans le fichier de paramètres préalablement chargé. Une fois le shapefile entièrement traité, donc la classe étudiée dans son intégralité, les polygones ainsi créés sont enregistrés dans un fichier shapefile en fonction du type de l'objet englobé.

**Il y a en sortie un shapefile de polygones par type d'objet.**

Lorsque le calcul se termine, QGis est automatiquement lancé et toutes les couches vectorielles créées sont chargées.

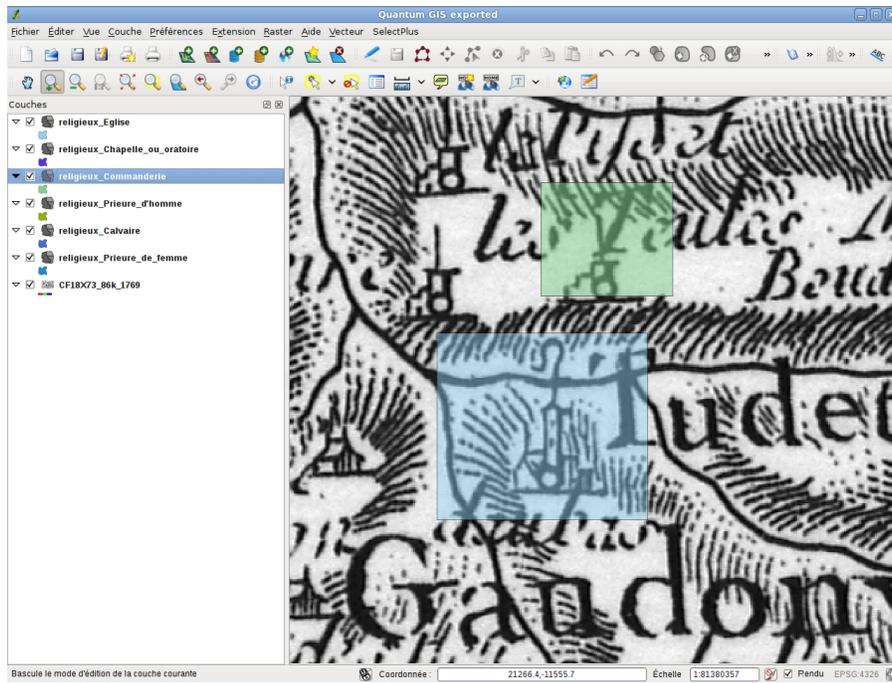


FIGURE 4 – Visualisation avec QGis des polygones englobants créés automatiquement.

### 0.3.2 Opération de recadrage

La figure 4 montre qu’il peut s’avérer nécessaire de modifier l’emprise des polygones afin d’ajuster le mieux possible le rectangle englobant au symbole. Lorsqu’un sommet est déplacé manuellement sous QGis, il est rare d’obtenir au final un rectangle dont les arrêtes sont parfaitement parallèles deux à deux (cf. figure 5). L’algorithme intègre donc une opération de recadrage permettant d’obtenir une géométrie rectangulaire définie comme le rectangle englobant minimal du polygone modifié (minimum bounding rectangle). Les shapefiles des polygones englobants sont écrasés et on ne récupère en sortie que les géométries ainsi modifiées.

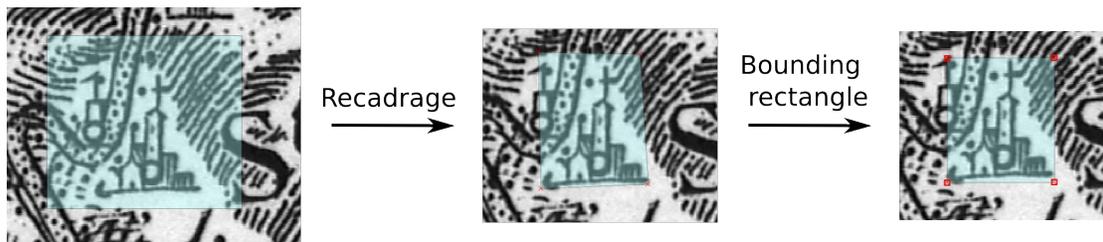


FIGURE 5 – Opération de rectangularisation après recadrage des polygones.

### 0.3.3 Extraction des vignettes

L’extraction des vignettes se fait automatiquement via l’utilisation de la bibliothèque Java JAI (Java Advanced Image <sup>1</sup>) en utilisant comme paramètres les paires de coordonnées (xmin, ymin) et (xmax, ymax) de chaque polygone. Chaque vignette est enregistrée dans un dossier en fonction de son type.

1. Site : <http://java.sun.com/javase/technologies/desktop/media/>

## 0.4 Données en sortie et conventions

### 0.4.1 Axes

Par convention, on choisit pour origine le coin supérieur gauche de l'image et un axe des ordonnées orienté vers le bas (attention, sous QGIS l'axe des y est orienté dans le sens contraire, les ordonnées étant alors négatives). Ce choix est conforme au système d'axes utilisé par JAI et aux conventions établies avec le LIP6.

### 0.4.2 Polygones

Les polygones rectifiés (rectangularisés) sont enregistrés selon une arborescence du type `/polygones/classeI/classeI_typeJ.shp`.

*Exemple* : les polygones englobants les objets de type « Église » sont enregistrés dans le shapefile `/polygones/Religieux/Religieux_Eglise.shp`

### 0.4.3 Vignettes

Les vignettes sont enregistrées selon une arborescence du type `/img/classeI/typeJ/typeJ_num.tif` avec num l'identifiant de l'image autoincrémenté à chaque création d'une image du même type typeJ. Exemple : si il y a deux vignettes d'église, elles auront comme chemin `/img/religieux/Eglise/Eglise_0.tif` et `/img/religieux/Eglise/Eglise_1.tif`.

### 0.4.4 Vérification

Un script Bash a été développé au COGIT et permet de créer une mosaïque de l'ensemble des vignettes présentent dans un dossier. Cette nouvelle image permet de détecter d'éventuelles erreurs de saisie (un écart annoté comme un hameau par exemple), et de vérifier que les vignettes collent bien au plus près les entités ponctuelles.

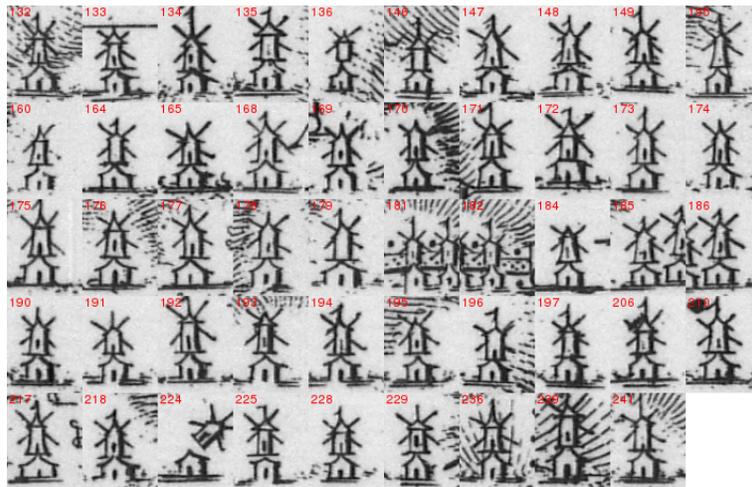
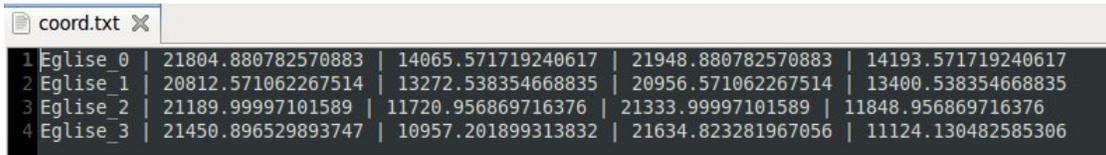


FIGURE 6 – Mosaïque de vignettes pour les moulins à vent de la feuille de Reims.

### 0.4.5 Fichier des coordonnées

Chaque image est fournie avec ses coordonnées dans l'image d'origine. Les coordonnées sont triées par type. Dans chaque répertoire /typeJ, on trouve donc un fichier coord.txt contenant les coordonnées de toutes les images du type typeJ. Chaque ligne de ce fichier correspond à une image et renseigne :

- Le nom de l'image
- Ses coordonnées, xmin, ymin, xmax et ymax avec les conventions définies plus haut concernant les axes et l'origine.



```
coord.txt X
1 Eglise_0 | 21804.880782570883 | 14065.571719240617 | 21948.880782570883 | 14193.571719240617
2 Eglise_1 | 20812.571062267514 | 13272.538354668835 | 20956.571062267514 | 13400.538354668835
3 Eglise_2 | 21189.99997101589 | 11720.956869716376 | 21333.99997101589 | 11848.956869716376
4 Eglise_3 | 21450.896529893747 | 10957.201899313832 | 21634.823281967056 | 11124.130482585306
```

FIGURE 7 – Exemple de fichier coord.txt (pour les églises ici).

## 0.5 Schéma récapitulatif

Le diagramme ci-dessous présente les différents processus mis en œuvre et leur enchaînement. Les cercles rouge barrés de noir représentent les moteurs du programme Java développé, l'autre opération (« Réajustement manuel des emprises ») demeurant manuelle. Les données en entrées sont en bleu, celle en sortie en vert et les intermédiaires utilisées par le programme en noir.

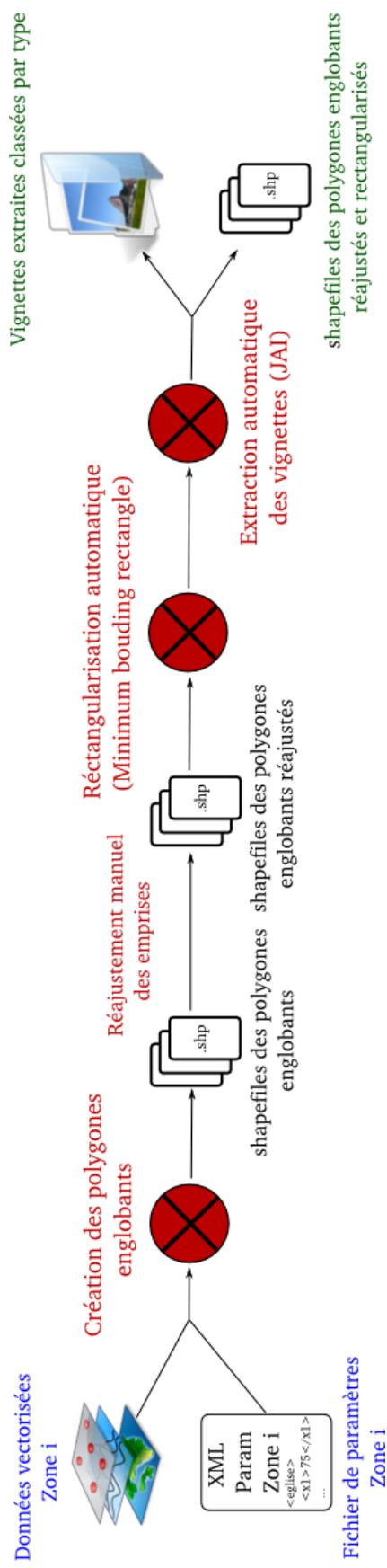


FIGURE 8 – Diagramme d'enchaînement des processus.

## 0.6 Manuel d'utilisation

On détaille dans cette section le fonctionnement de l'outil développé. Le livrable comporte les éléments suivants :

- un répertoire « config » contenant les quatre fichiers de paramètres (un par zone),
- un fichier de démarrage Windows « geopeuple\_windows.bat » ,
- un fichier de démarrage Linux « geopeuple\_linux.sh »,
- un fichier contenant l'application « geopeupleimageextraction0.1.jar».

Une machine virtuelle Java est nécessaire pour lancer le programme. On peut en télécharger une sur le site d'Oracle<sup>2</sup>. Pour lancer l'application il suffit de double-cliquer sur l'un des fichiers de démarrage (selon qu'on utilise un environnement Linux ou Windows).

L'interface suivante s'affiche alors :



FIGURE 9 – Interface de lancement de l'extraction de vignettes.

L'utilisateur a deux possibilités :

- lancer le calcul de polygones englobants,
- lancer la procédure d'extraction de vignette.

### 0.6.1 Calcul des polygones

La fenêtre suivante s'ouvre alors :

---

2. Site d'Oracle : <http://java.com/fr/download/index.jsp>

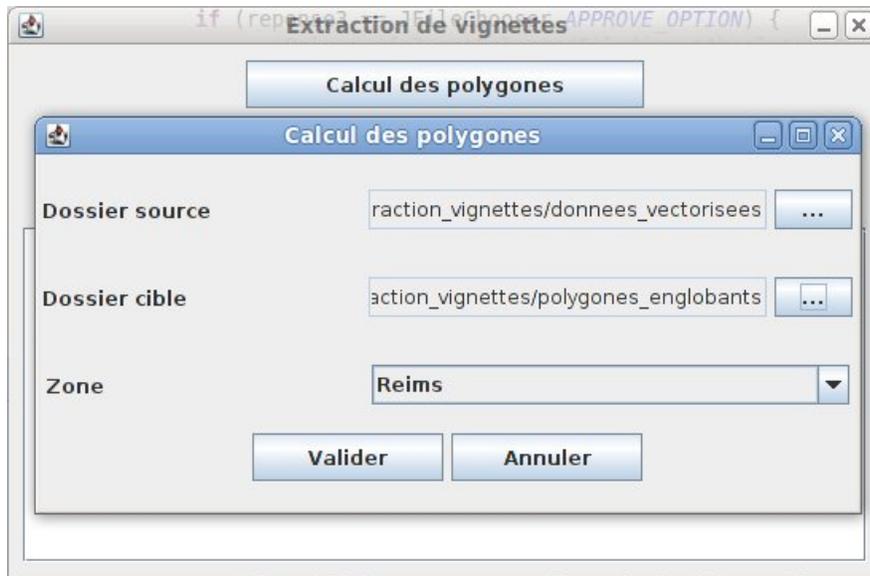


FIGURE 10 – Interface de calcul des polygones englobants.

L'utilisateur doit y renseigner :

- le répertoire source qui contient les fichiers au formats ESRI shapefile des données vectorisées. Le dossier peut aussi bien contenir un seul fichier que toute la base (dans ce cas, seuls les objets dont les dimensions des polygones englobants optimaux sont définis dans le fichier de paramètres seront pris en compte),
- le répertoire cible où seront enregistrés les shapefiles issus du calcul,
- la zone étudiée (Reims, Agen, Grenoble ou SaintMalo).

### 0.6.2 Fichier de paramètres

Le programme chargera directement le bon fichier de paramètres en fonction de la zone choisie. Ces fichiers XML se trouvent dans le répertoire config et se présente sous la forme suivante :

```
<ParametersCollection>
  <list>
    <DimensionParameters>
      <type>Eglise</type>
      <x1>77</x1>
      <x2>67</x2>
      <y1>100</y1>
      <y2>28</y2>
    </DimensionParameters>
    [...]
  </list>
</ParametersCollection>
```

On retrouve bien, pour chaque type, les dimensions x1, x2, y1 et y2 définies précédemment dans ce rapport.

L'utilisateur peut si il le souhaite ajuster ces valeurs, ou rajouter un nouveau type (car seuls les types présents dans le fichier de paramètres sont pris en compte par le programme pour le calcul de polygones et donc l'extraction des vignettes). Pour rajouter un type *typeJ* dont les dimensions du polygone englobant optimal sont a, b, c et d, il suffit de rajouter avant la balise </list>, les lignes suivantes :

```
<DimensionParameters>
  <type>typeJ</type>
  <x1>a</x1>
  <x2>b</x2>
  <y1>c</y1>
  <y2>d</y2>
</DimensionParameters>
```

### 0.6.3 Extraction des vignettes

L'utilisateur peut lancer la procédure d'extraction en utilisant les polygones qui viennent d'être calculés si la procédure de construction des polygones a été lancée précédemment (et que l'application n'a pas été fermée depuis).

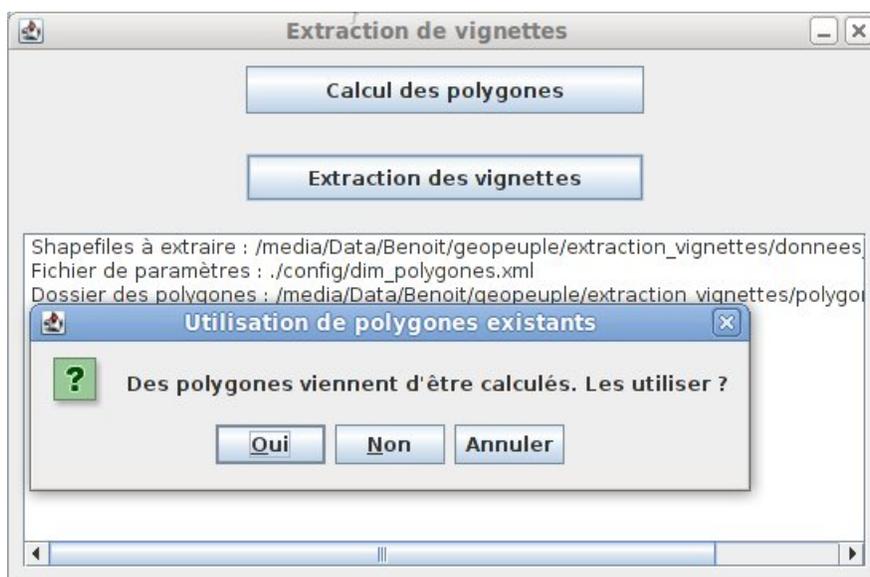


FIGURE 11 – Utilisation des polygones précédemment calculés.

La fenêtre suivante s'ouvre alors :

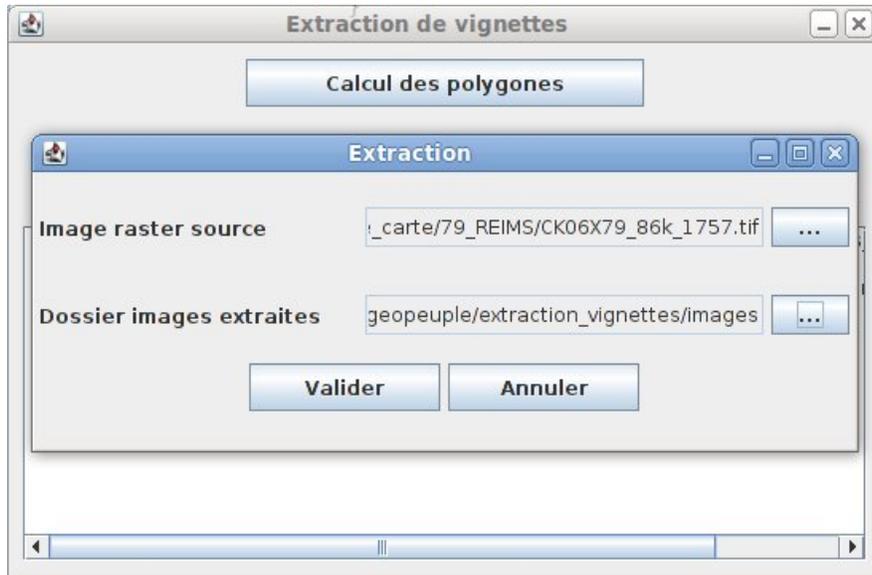


FIGURE 12 – Extraction des vignettes en utilisant les polygones précédemment calculés.

Il faut alors indiquer le fichier raster à découper ainsi que le dossier dans lequel il souhaite enregistrer les images qui vont être produites.

L'opérateur peut également lancer cette procédure indépendamment d'un éventuel calcul de polygones précédent en renseignant le chemin du répertoire contenant les shapefiles des polygones (qui auront par exemple été calculés la veille).



FIGURE 13 – Extraction des vignettes en utilisant d'autres polygones.

L'utilisateur définit alors de la même façon le fichier raster à découper ainsi que le dossier dans lequel il souhaite enregistrer les images qui vont être produites, mais également le répertoire contenant les polygones à utiliser.

La figure 14 ci-dessous montre un exemple de résultat d'extraction.

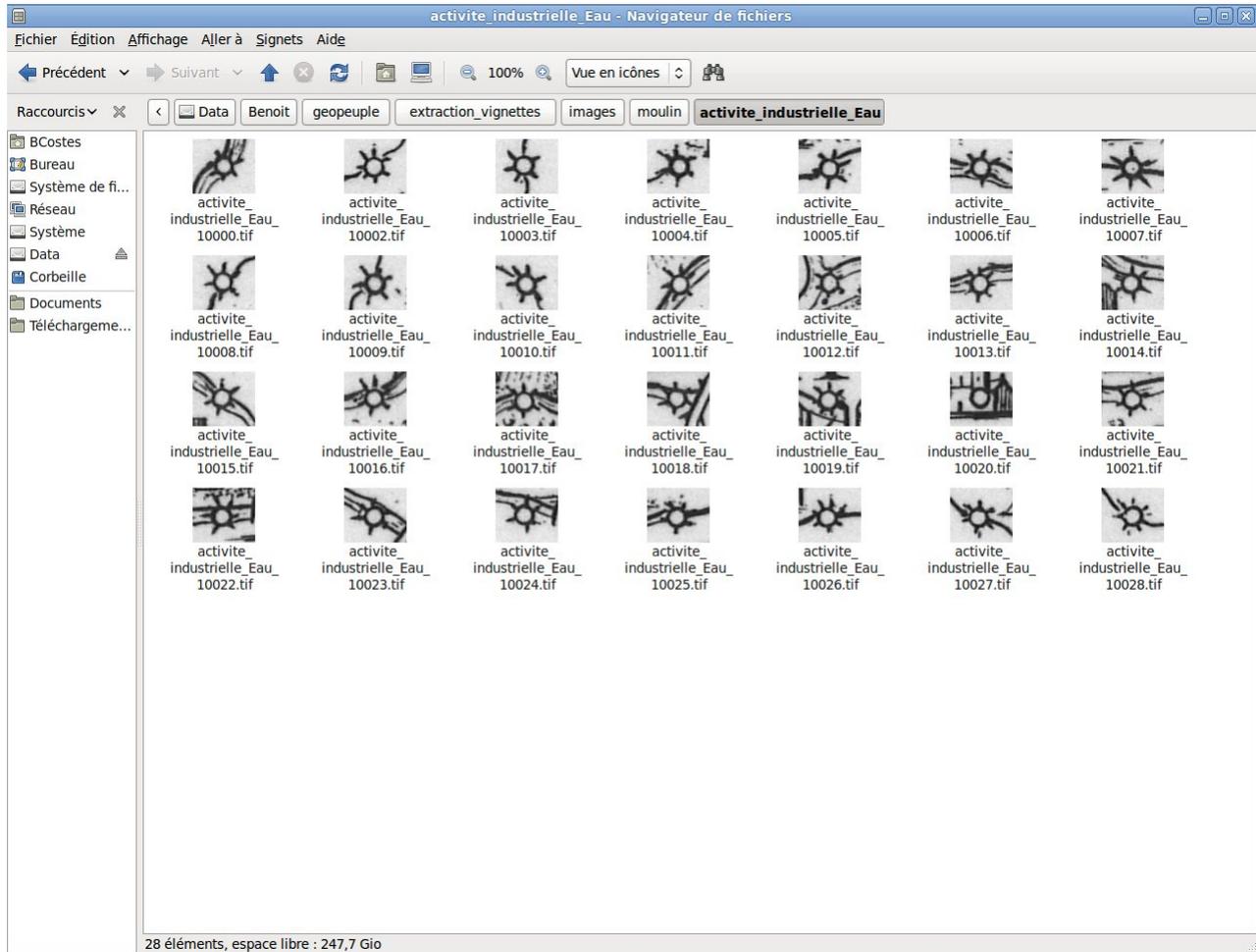


FIGURE 14 – Extraction terminée pour les moulins à eau.