ENSG Géomatique
ÉCOLE NATIONALE DES SCIENCES GÉOGRAPHIQUES

Ecole Nationale des Sciences Géographiques

UNIVERSITÉ PARIS-EST

Université Paris-Est

-PROJET INFORMATIQUE -
Rapport programmeur

Mastère spécialisé ® Photogrammétrie, Positionnement, Mesure de Déformations

Développement d'une interface graphique pour améliorer des classifications de manière interactive

ENSG Géomatique
ÉCOLE NATIONALE DES SCIENCES GÉOGRAPHIQUES

CHUPIN Clémence

Janvier-Février 2018

☒ Non confidentiel ☐ Confidentiel IGN ☐ Confidentiel Industrie ☐ Jusqu'au ...

Ecole Nationale des Sciences Géographiques
6-8 Avenue Blaise Pascal - Cité Descartes - 77420 Champs-sur-Marne
Téléphone 01 64 15 31 00 Télécopie 01 64 15 31 07

Table des matières

Introduction

Dans le cadre des projets informatiques des étudiants du Mastère Spécialisé PPMD de l'ENSG, le laboratoire MATIS de l'IGN a proposé un sujet se rattachant à la problématique des modèles 3D urbains. Ce sujet s'inscrit dans le cadre de la thèse de M.ENNAFII Oussama, intitulée "Evaluation and selection of 3D city modelling techniques". Pour détecter et caractériser les erreurs des maquettes 3D urbaines, une méthode d'auto-qualification par classification supervisée a été mise en place. Pour permettre la transition vers une classification active,le projet propose de mettre en place une interface graphique pour interagir avec un utilisateur.

Ce rapport programmeur a pour objectif de décrire dans le détaille le code et les données du programme. Il est complété par le rapport de programmation et le rapport utilisateur.

Chapitre 1 Structure générale

1.1 Configuration requise

Le langage de programmation utilisé pour le code est le Python. La partie interface graphique utilise la bibliothèque Qt, et c'est la version 5 de PyQt qui a été utilisée.

D'autres packages de Python sont nécessaires au bon fonctionnement du programme :

- gdal, permettant la gestion données géospatiales vectorielles ou raster ;
- pyshp, permettant l'extraction et la manipulation de Shapefile ;
- qimage2ndarray, une extension permettant la conversion entre les QImages et numpy.ndarray.

Le programme est multi-plateforme : il peut donc être ouvert sous Linux, Windows ou MAc OS.

1.2 Données et formalisme

Le programme considère plusieurs données en entrée, sélectionnées par l'utilisateur via une première interface :

Données	Type de donnés	Correspondance dans le code
Classes d'erreurs possibles	Fichier .CSV	Variable de type dictionnaire
Résultats de l'auto-qualification	Fichier .CSV	Liste de tuples
Géométrie des entités	Dossier de .SHP	Attribut *geometry* de la classe Bâtiment
Orthophoto	Fichier .TIFF	Tuple (résolution + référence) + Matrice

Figure 1.1 – Données en entrée

De même, le formalisme des données en sortie est imposé :

Données	Type de donnés	Correspondance dans le code
Résultats de l'interaction	Fichier .CSV	Liste de tuples

Figure 1.2 – Données en sortie

Chapitre 2 Détail du code

Pour faciliter l'écriture et la compréhension du code, on utilise la méthode Modèle/Vue/Contrôleur. Dans cette section, on détaillera donc les différentes fonctions implémentées en suivant ce modèle.

2.1 Partie Vue

La partie Vue contient le code relatif à l'interface graphique. Après la réalisation des trois interfaces graphiques avec QtDesigner, l'utilitaire PyUic5 a permis de transcrire le code C++ en langage Python.

Trois fichiers ont alors été créés et définissent les trois classes :

→ La classe *Ui_InterfacePrincipale* du fichier *classificationActive.py* comporte le code de l'interface principale. Elle définit les caractéristiques des boutons, des labels de texte et de la fenêtre graphique dans une première méthode *setupui* : ce sont les attributs de l'interface. Une méthode *retranslateUi* permet de mettre à jour le texte des attributs.

→ La classe *Ui_ChargementFichiers* du fichier *chargementFichiers.py* comporte le code de l'interface de chargement. Elle dispose elle aussi des deux méthodes *setupui* et *retranslateUi*. Pour permettre l'affichage des différentes stratégies dans le menu déroulant, on utilise la variable globale STRATEGIES définie dans le fichier *strategy.py*. Ainsi, ce menu sera mis à jours automatiquement en cas d'implémentation d'une nouvelle stratégie.

→ La classe *Ui_ChoixClasse* du fichier *choixClasse.py* comporte le code de l'interface de sélection d'une nouvelle classe. Elle dispose des méthodes *setupui* et *retranslateUi*.

En cas de modification de l'interface, ces fichiers sont réécrits, mais cela n'impactera pas le code du programme principal. Une seule modification doit être apportée à ces codes générés automatiquement : il faut modifier le chemin de l'image du point d'interrogation. En effet, ce chemin est contenue dans un fichier ressource en C++, qui n'est pas traduisible simplement en Python avec PyUic5.

2.2 Partie Modèle

La partie Modèle contient organise le formalisme des données. Ainsi, trois nouveaux fichiers viennent définit trois classes : la classe *Building*, la classe *Background* et la classe *Strategy*.

2.2.1 La classe *Building*

La classe *Building* est codée dans le fichier *building.py*. Un objet de type *Building* est défini par 4 attributs : identity (identifiant de l'objet), geometry, classe et probability. Deux méthodes lui sont associées :

— *get_points* retourne la liste de tous les sommets de chaque géométrie de l'entité. Ainsi, si une entité est composée de 3 triangles, la liste de sortie contiendra 9 points. Chaque point est codé par un couple de coordonnées (x,y).

— *get_bounding_box* parcourt l'ensemble des coordonnées x et y d'une liste et retourne les valeurs maximales et minimales de ces deux paramètres. Cela permet de définir la fenêtre d'emprise d'une entité.

2.3 Partie Contrôleur

Table des figures