



ÉCOLE NATIONALE
DES SCIENCES
GÉOGRAPHIQUES

École Nationale des
Sciences Géographiques



Open Street Map



Université Gustave Eiffel



INSTITUT NATIONAL
DE L'INFORMATION
GÉOGRAPHIQUE
ET FORESTIÈRE

Institut Géographique
National



Panoramax

Rapport d'analyse - Modélisation et implémentation d'une BD PANNEAU à partir de Panoramax

BERGERET Clovis, GIRARDIN Claire, GIUDICELLI Vincent, OLLIVIER Victorien,
ROUILLARD Mathis, élèves ingénieur·e·s à l'ENSG Géomatique, spécialité TSI

23 Avril 2024

Sommaire

Glossaire et sigles utiles

Introduction	1
1 Contexte du projet	2
1.1 Contexte pédagogique	2
1.2 Panoramax, un <i>Street View</i> collaboratif	2
2 Objectifs de l'étude	3
2.1 Objectifs de l'étude : modéliser et implémenter une BD PANNEAU	3
2.2 Les contraintes	4
2.2.1 Qualité de la donnée ouverte	4
2.2.2 Contraintes intrinsèques à un projet informatique	4
2.3 Recueil du besoin et acteurs impliqués	5
2.4 Données utilisées	6
3 Analyse fonctionnelle - solution proposée	7
3.1 Définition d'une chaîne de traitement	7
3.1.1 Chaîne de traitement finale retenue	7
3.1.2 Hypothèses émises	8
3.2 Mise en oeuvre du <i>workflow</i>	8
3.2.1 Enrichissement des panneaux détectés	8
3.2.2 Création d'une base de données intermédiaire	12
3.2.3 Intersecter les panneaux - Utilisation de méthodes inspirées de la photogrammétrie	13
3.3 Évaluation de la qualité	17
3.3.1 Critères de qualités minimum sur les données	18
3.3.2 Précision de l'algorithme de détection de centre et de hauteur	19
3.3.3 Précision de l'algorithme de localisation des panneaux	21
4 Étude technique	23
4.1 Langages, techniques et outils utilisés	23
4.2 Gestion de versions, intégration et déploiement continu	23
4.2.1 Git	23
4.2.2 Docker	23
4.3 Architecture technique et physique de la solution	24
5 Réalisation et suivi du projet	25
5.1 Phasage du projet	25
5.2 Travail réalisé	27
5.3 Problèmes rencontrés	27
5.4 Évolutions possibles	28
5.4.1 Évolutions de la chaîne de traitement	28

0. SOMMAIRE

5.4.2 Autres évolutions	28
Conclusion	29

Table des figures

2.1	Imagetts de détection de panneaux	3
2.2	Un même panneau peut être présent sur plusieurs photos d'une ou plusieurs séquences	4
3.1	Diagramme de séquence du <i>workflow</i> global	8
3.2	Association forme - code pour le dictionnaire	9
3.3	Diagramme d'activité modélisant le calcul de centre de panneau	10
3.4	Schémas des cas possibles lors du calcul de la hauteur	11
3.5	Schéma montrant la récupération des sommets lors d'une détection de contour intérieur	12
3.6	Diagramme d'activité modélisant le calcul de hauteur de panneau	13
3.7	Exemple de calculs d'orientation relative des panneaux dans les imagettes	14
3.8	Diagramme de classe représentant le modèle de données de la BD PANNEAU	15
3.9	Lecture du gisement dans une prise de vue	16
3.10	Schéma du calcul largeur/angle	17
3.11	Calcul du facteur d'échelle	17
3.12	Critères de compatibilités	18
3.13	Explication second critère	19
3.14	Exemple de clusterisations compatibles et non compatibles	20
3.15	Algorithme de clusterisation de matrice de compatibilité	20
3.16	Imagetts n'ayant pu être traitées car les formes détectées sont trop petites	21
3.17	En haut, de gauche à droite : bonnes valeurs, centre ok, hauteur ok ; en bas, de gauche à droite : seulement le centre, mauvaises valeurs et résultats aberrants	22
3.18	Résultats sur le jeu test	22
4.1	Diagramme de composants	24
5.1	Rétroplanning effectué en début de projet	25
5.2	Diagramme de Gantt effectué au début du projet	26
5.3	Matrice des risques de la deuxième semaine du projet	26
5.4	Matrice des risques de la dernière semaine du projet	27

Glossaire et sigles utiles

ENSG École Nationale des Sciences Géographiques

IGN Institut National de l'Information Géographique et Forestière

UML Unified Modeling Language

TSI Technologies des Systèmes d'Information

OSM OpenStreetMap

SQL Structured Query Langage

EXIF EXchangeable Image File format

SI Système d'Information

Introduction

Comme défini sur le site Panoramax.fr, "Panoramax est une ressource numérique permettant la mise en commun et l'exploitation de photos de terrain. Toute personne peut photographier des lieux visibles depuis la voie publique afin d'alimenter la base de données de Panoramax. Ces données sont ensuite librement accessibles et réutilisables". Startup d'Etat sous la tutelle de l'IGN et d'OSM, Panoramax a proposé un sujet de projet de fin d'année à la promotion TSI de l'ENSG-Géomatique 2023/2024. Ce projet consiste en la modélisation et création d'une "BD PANNEAU". Celle-ci est basée sur une extraction des informations de localisation des panneaux présents sur les photos disponibles sur Panoramax. Récupérer les photos, effectuer les traitements nécessaires pour les géolocaliser et documenter les précisions obtenues implique divers domaines de la géomatique, et particulièrement l'application de processus inspirés de la photogrammétrie.

Ce projet s'inscrit parfaitement dans le contexte d'une volonté globale de mutualiser les ressources informatiques à l'instar des données, volonté trouvant son vivier de contributeurs dans la communauté *open source*.

Ce rapport d'analyse a pour vocation de décrire le contexte et les moyens mis en œuvre pour la réalisation du projet, ainsi que de détailler la conceptualisation de la solution logicielle modélisée et implémentée lors du projet.

Vous trouverez ci-dessous le lien du dépôt *Github*.

https://github.com/VGiudicelli1/TSI_panoramax

1. Contexte du projet

1.1 Contexte pédagogique

Le projet Panoramax s'intègre dans le cadre du projet de fin d'année de la filière Technologies des Systèmes d'Information (TSI) à l'École Nationale des Sciences Géographiques (ENSG). Son objectif est de mettre en pratique les compétences techniques et de gestion de projet acquises au cours du cursus, que ce soit dans le tronc commun ou lors de la filière. Contrairement aux projets antérieurs, qui étaient davantage encadrés, celui-ci vise à favoriser l'autonomie des étudiants.

La présentation des projets a eu lieu le jeudi 21 mars 2024. La date de clôture, fixée pour la soutenance finale, est le 30 avril 2024. Une première version de ce rapport était attendue pour le 23 avril 2024.

1.2 Panoramax, un *Street View* collaboratif

Le projet choisi a été celui de la start-up d'État Panoramax, présentée en introduction. L'ancrage du concept de Panoramax dans le domaine de la géomatique s'est solidifié et s'est étendu à divers contextes d'utilisation, allant des institutions administratives aux utilisateurs individuels. Cependant, la vision de la start-up va au-delà de la simple visualisation des données recensées sur son site web. En effet, Panoramax aspire à valoriser pleinement l'utilisation de ces données, mettant en avant la diversité des usages potentiels des photographies *open source* collectées par ses contributeurs.

Cela signifie que Panoramax ne se limite pas à une plateforme de visualisation, mais cherche plutôt à créer un écosystème où les données peuvent être exploitées de manière gratuite et innovante. Cette approche met en exergue la volonté de la start-up de favoriser l'émergence de nouvelles utilisations et de nouvelles applications des données géospatiales.

Le projet à réaliser par les étudiants s'inscrit dans une volonté de Panoramax de réaliser une BD PANNEAU, idéalement rattachée à la BD TOPO®.

2. Objectifs de l'étude

2.1 Objectifs de l'étude : modéliser et implémenter une BD PANNEAU

L'objectif de l'étude est de modéliser et d'implémenter une base de données des panneaux en France, dénommée dans la suite BD PANNEAU.

Les photographies sont riches en informations. Beaucoup de photos capturent des panneaux de signalisation. Certaines personnes de Panoramax ont déjà procédé à une détection des panneaux de signalisation dans les photographies via un algorithme d'apprentissage profond. Cela a permis de constituer une base de données d'image représentant chacune un panneau détecté dans une photo, comme il est montré ci-dessous (fig 2.1). Dans la suite, les images détectées par cet algorithme seront appelées **imagettes** (nom provenant de la taille des images en question, assez petite par rapport aux photos d'origines). Ces imagettes sont classées en fonction du type du panneau détecté.



FIGURE 2.1 : Imagettes de détection de panneaux

Les imagettes sont localisées puisque les appareils qui les ont prises le sont aussi au moment de l'acquisition de la photographie. Ainsi, un panneau peut être photographié, puis détecté à l'occasion de différentes acquisitions.

Comme on peut le voir sur la figure 2.2, des acquisitions de plusieurs personnes ou appareils peuvent détecter un même panneau. On voit aussi que les photos d'un même panneau sont à des endroits bien espacés, de l'ordre de la dizaine de mètres.

Notre objectif est de localiser chaque panneau, ce qui implique de les dédoublonner et de savoir les reconnaître sur les différentes acquisitions. Cela nous permettra de construire la BD PANNEAU, une base de données des panneaux avec une précision suffisante pour que la diffusion de l'information soit légitime.



FIGURE 2.2 : Un même panneau peut être présent sur plusieurs photos d'une ou plusieurs séquences

2.2 Les contraintes

2.2.1 Qualité de la donnée ouverte

Nous avons rapidement identifié une contrainte majeure à laquelle nous allions être confrontés : la qualité des acquisitions. Tout d'abord, la qualité intrinsèque de la photo elle-même est cruciale. Une photo floue, sur ou sous-exposée, ou de faible résolution, rend son utilisation difficile, voire impossible. Ces défauts sont perceptibles à l'œil nu.

Cependant, d'autres contraintes liées aux acquisitions sont également à prendre en compte. Par exemple, l'absence de paramètres extrinsèques, voire parfois intrinsèques, de l'appareil peut rendre le traitement des photos complexe, voire impossible, par des méthodes photogrammétriques. De plus, les métadonnées disponibles sur le site de Panoramax sont rarement toutes renseignées par les divers appareils utilisés par les contributeurs, rendant alors les acquisitions inutilisables.

En outre, lorsque les photos sont prises sans positionnement précis (RTK par exemple), ce qui est souvent le cas, la géolocalisation de l'appareil présente déjà une marge d'erreur de quelques mètres. Cette imprécision peut poser des défis supplémentaires lors de la géolocalisation des panneaux détectés.

2.2.2 Contraintes intrinsèques à un projet informatique

Lors de projets informatiques, plusieurs contraintes habituelles doivent être prises en compte. Tout d'abord, la contrainte de temps demeure toujours pressante. L'efficacité de sa gestion est essentielle, d'où la nécessité de prioriser, notamment par le biais d'un recueil du besoin.

Il est également crucial de considérer la contrainte de l'*open source* et de la maintenabilité du code pour l'avenir. La décision de rendre nos implémentations disponibles et publiques implique de suivre les bonnes pratiques de l'*open source*, telles que la rédaction de code clair, l'inclusion de commentaires en anglais, une attention particulière à la lisibilité du code, ainsi que la création de documentation détaillée pour l'utilisation et le développement de chaque partie du logiciel.

2. Objectifs de l'étude

La question de la maintenabilité soulève également la réflexion sur la généricité des solutions proposées. Compte tenu de la probabilité que les codes *open source* soient réutilisés, modifiés et améliorés, il est primordial de les concevoir de sorte à ce qu'ils puissent être intégrés facilement dans l'écosystème informatique de Panoramax. Par exemple, il convient de se demander si les traitements appliqués sur nos jeux de données tests doivent être automatiquement appliqués dès l'ajout de nouvelles photos sur le site, ou s'il est préférable de laisser les développeurs les lancer manuellement. Ces questions trouvent leurs réponses à la fois dans l'analyse fonctionnelle détaillée dans la suite, ainsi que dans les contraintes de temps. En effet, l'intégration des chaînes de traitement développées à Panoramax nécessite qu'elles soient suffisamment robustes pour être intégrées sans compromettre la qualité du Système d'Information (SI), ce qui demande d'y dédier un temps important.

2.3 Recueil du besoin et acteurs impliqués

Nous avons défini le recueil du besoin suivant :

- Modéliser la BD PANNEAU via l'élaboration d'un modèle conceptuel de données en regard des usages considérés.
- Dédoublonner les détections
- Élaborer une méthode de géolocalisation des panneaux
- Géolocaliser les panneaux le plus précisément possible
- Implémenter la BD PANNEAU
- Créer un Webservice exposant ces données

Ce recueil fédère plusieurs acteurs autour du projet, aux rôles différents dans la concrétisation de ces objectifs. Ces acteurs sont les suivants :

Les commanditaires de Panoramax ont mandaté le développement de la solution. Leur rôle principal est de définir les objectifs du projet, de fournir des ressources nécessaires, et de valider les livrables tout au long du processus de développement. Ils sont également responsables de la communication des attentes et des exigences du projet aux développeurs, ainsi que de la prise de décisions pour garantir la réussite du projet.

Notre équipe, composée de développeur·euse·s et de responsables de la gestion de projet, est chargée de la conception, du développement et de la mise en œuvre du projet. Nous sommes responsables de traduire les besoins et les exigences en solutions techniques, de planifier et d'organiser les différentes phases du projet, et de coordonner les efforts de développement.

ENSG : Les enseignants, tels que Victor Coindet et Julien Levesy, représentent l'ENSG dans la mesure où ils évalueront les compétences mises en œuvre lors du projet.

Les contributeurs de Panoramax sont les individus ou organisations qui contribuent à la collecte et à la mise à jour des données géospatiales sur la plateforme. Ce sont donc eux qui sont à l'origine des données que nous avons utilisées.

Les utilisateurs de la BD PANNEAU : nous avons rencontré de potentiels utilisateurs de la BD PANNEAU. Par exemple, nous avons rencontré Romain Veldeman, gendarme de profession, qui nous a présenté les intérêts qu'il trouverait dans une telle base de données. L'Institut National de l'Information Géographique et Forestière (IGN) consiste également en un utilisateur, puisque les informations de la BD PANNEAU pourraient enrichir les métadonnées des routes de la BD TOPO® (en spécifiant les vitesses par exemple). Elle pourrait ainsi être utilisée pour la création d'un service public d'itinéraire.

2.4 Données utilisées

Naturellement, la donnée principale d'un service tel que Panoramax est de la donnée photographique. En naviguant à travers l'application, on peut visionner et télécharger les photos hébergées sur le site. À chaque photo sont associés plusieurs éléments, tels que le nom du contributeur qui l'a prise, la date, la géolocalisation, le type d'image (photo plane ou 360°), et d'autres données présentes sous forme de tags EXchangeable Image File format (EXIF). Chaque photo est consultable via l'API Panoramax, et l'URL correspondant à celle-ci est composée d'un identifiant d'image ainsi que d'un identifiant de séquence, qui est un trajet de plusieurs photos prises en une seule acquisition.

Il est intéressant de noter que Panoramax, en raison de son caractère communautaire, ne pose que quelques critères élémentaires concernant les contributions :

- L'image doit être une photo
- La photo doit être géolocalisée, sans précision minimum
- La photo doit être datée

Il est donc courant que plusieurs séquences se trouvent au même endroit, passent sur un même tronçon de route. Ainsi, il est possible pour un même panneau d'être aperçu par plusieurs séquences, et donc d'apparaître dans nos données sous plusieurs résolutions (image ou positionnement) différentes.

Lorsqu'on enregistre les images en local, on garde les tags EXIF et il est alors possible de les lire avec une bibliothèque telle que *exiftool*.

Les commanditaires nous ont également fourni les imagettes (images de panneaux découpées issues des images de Panoramax) de tous les panneaux détectés sur la ville de Strasbourg. En effet, la qualité des acquisitions à Strasbourg est bonne, car effectuée par des professionnels, et la ville a également une esquisse de BD PANNEAU, qui pourrait nous être utile pour comparer nos résultats. Cependant, il s'agit surtout de la qualité de localisation des images qui est exemplaire. La qualité des photographies est très variable. De nombreux panneaux détectés sont flous et la luminosité varie beaucoup. Cela contraint donc énormément la reconnaissance des panneaux. Ces imagettes, puisqu'elles sont issues des images originales, conservent certains tags EXIF, comme la localisation, ce qui permettra dans la partie suivante de les associer.

3. Analyse fonctionnelle - solution proposée

Notations

angles : degrés

distances (et tailles) : mètres

Note : les angles sont donnés dans le sens horaire par rapport au nord. On suppose travailler dans une projection qui induise localement des déformations négligeables

3.1 Définition d'une chaîne de traitement

3.1.1 Chaîne de traitement finale retenue

Il nous est rapidement apparu que le cœur du sujet résidait en la concrétisation d'une chaîne de traitements passant des *input*, les imagettes et images de Panoramax, aux *output*, les localisations des panneaux.

Nous avons défini plusieurs étapes afin de modéliser cette chaîne. Voici ce que nous avons retenu :

1. On récupère des imagettes
2. On récupère la localisation des images de base avec l'API *Search*
3. On récupère les informations de l'image originale
4. On insère dans la base de données intermédiaire l'image, l'imagette et leurs informations
5. On détermine le centre du panneau
6. On détermine la hauteur du panneau
7. On détermine l'orientation du panneau
8. On remplit la base de données intermédiaire avec les informations explicitées et calculées
9. On calcule la position des panneaux et on les dédouble
10. On complète la base de données BD PANNEAU

Cette chaîne de traitement se modélise particulièrement bien avec un diagramme de séquence (voir figure 3.1).

Comme on peut le voir, une grande partie de la chaîne de traitement est effectuée sur chaque imagette, représentée par la boucle "*onCroppedSign*". Il semble important de préciser que ce diagramme, qui reflète le processus général, reste simplificateur quant à l'enchaînement des actions.

3. Analyse fonctionnelle - solution proposée

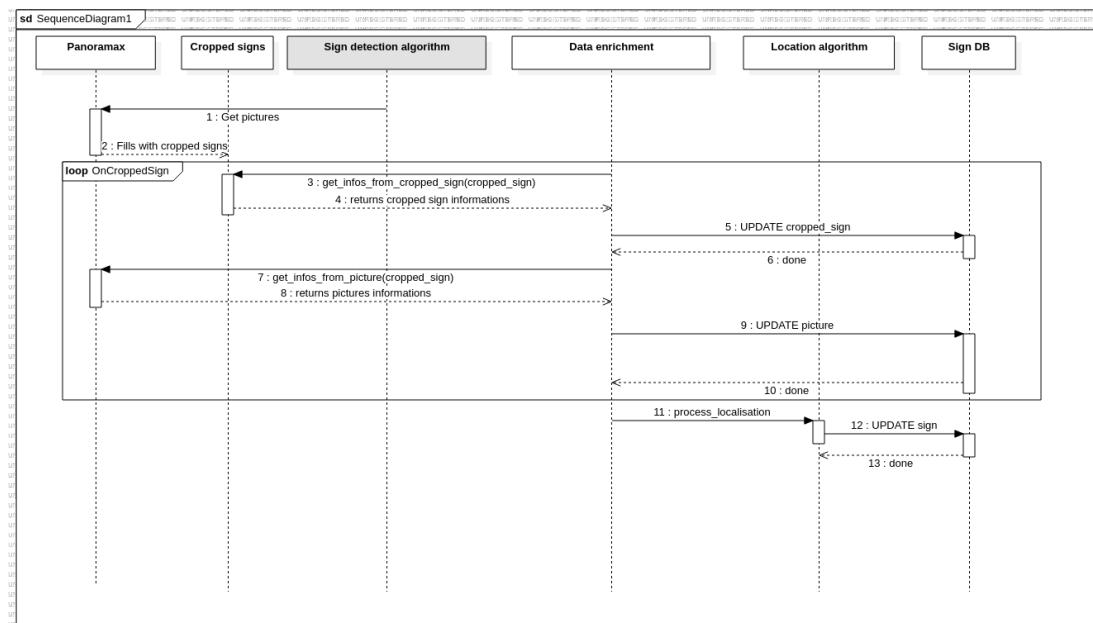


FIGURE 3.1 : Diagramme de séquence du *workflow* global

3.1.2 Hypothèses émises

Plusieurs hypothèses ont été émises puis abandonnées lors de la modélisation de ce *workflow*. Par exemple, nous avons envisagé de détecter automatiquement les contours du panneau, peu importe sa forme et sans dictionnaire. Très vite, nous nous sommes rendu compte que cela rendait l'algorithme moins performant que si nous pouvions chercher précisément une forme sur une image donnée. Implémenter la seconde hypothèse nous a permis de savoir si le panneau reconnu avait la forme escompté ou non, auquel cas nous pouvons l'exclure. Ainsi, nous avons abandonné l'idée de détecter automatiquement les contours et avons mis en place un dictionnaire.

Nous avions initialement fait le choix d'appliquer les calculs d'intersections sur les images consécutives d'une même séquence - un série de photos prises par un même appareil, dans un intervalle de temps réduit (30 minutes maximum entre deux photos), et dans une même zone géographique (distance entre deux prises de vue limitée à 50m) - mais celles-ci se sont avérées trop espacées en distance pour avoir suffisamment de redondance pour que les algorithmes de photogrammétrie soient efficaces. Nous avons donc fait le choix d'un calcul d'intersection sur l'ensemble des détections - qui pourra être par la suite appliquée à un tiling géographique - afin d'avoir suffisamment de redondance. Ce calcul s'inspire des principes de photogrammétrie, tout en s'adaptant aux contraintes de nos données.

Nous avions espéré pouvoir travailler avec toutes les images de la base de données, mais nous nous sommes vite rendu compte de l'impossibilité d'obtenir des mesures d'angles - qui sont à la base des calculs photogrammétriques d'intersection - sur la plupart des prises de vues, à cause de l'absence d'informations propres à la caméra telles que l'angle de vue, les déformations, et les traitements géométriques appliqués à chaque cliché. Nous avons donc choisi de n'utiliser que des images 360°, dans lesquelles ces mesures d'angles sont réalisables. De plus, ces images nécessitent un équipement plus important, sont souvent réalisées par des professionnels, et sont donc de qualité plus constante (à l'instar des acquisitions de la ville de Strasbourg sur lesquelles nous nous sommes basés).

3.2 Mise en oeuvre du *workflow*

3.2.1 Enrichissement des panneaux détectés

Récupérer des informations existantes

A l'aide des tags EXIF de l'imagette, nous pouvons récupérer plusieurs informations intéressantes. Tout d'abord les coordonnées de la photo de laquelle provient l'imagette. Grâce à l'API *Search*, nous pouvons donc récupérer cette photo par la suite. Sont aussi stockées les coordonnées du coin en haut à gauche de l'imagette (l'origine du repère) au sein de l'image originale, ainsi que sa taille (hauteur et largeur).

Pour automatiser cela, nous avons implémenté un script Python.

Calculer de nouvelles informations

Comme dit précédemment, nous avions prévu de détecter la **forme du panneau** à l'aide d'algorithmes de reconnaissance de forme. Cependant, il s'est avéré que la reconnaissance, en raison de la qualité fortement variable des images, était souvent mauvaise. Nous avons donc créé un dictionnaire qui, à chaque type de panneau (par rapport à son nom dans la nomenclature officielle), associe un code représentant sa forme. Le code associant une forme à un chiffre est le suivant.

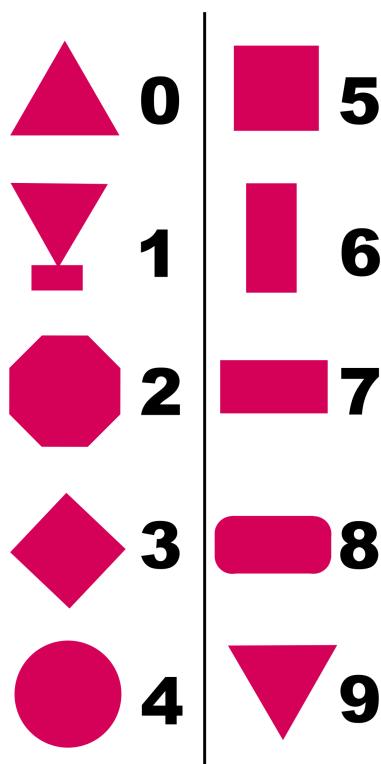


FIGURE 3.2 : Association forme - code pour le dictionnaire

Comme indiqué sur la figure 3.2, il y a plusieurs chiffres associés à des formes relativement similaires. Nous n'utilisons actuellement pas la différence, par exemple, entre un code 5, 6 ou 7 (rectangles configurés différemment). Cependant, nous avons envisagé que ces dissociations pourraient être utiles à l'avenir, par exemple lorsque la détection des panneaux d'agglomération sera plus aboutie. Le choix d'un dictionnaire a été effectué car le nombre de panneaux différents est, bien que significatif, limité. De cette manière, nous pouvons gagner en robustesse du code. L'algorithme vérifie que la forme trouvée dans l'imagette correspond bien à ce qui est recherché (afin de déterminer le centre et la hauteur par la suite par exemple).

Ensuite, nous avons procédé au calcul du **centre du panneau**. Pour cela, nous avons effectué une détection des contours de chaque imagette. Pour une imagette, nous avons appliqué un filtre de

3. Analyse fonctionnelle - solution proposée

Canny¹. Nous avons ensuite recherché le plus grand contour fermé de l'image. Avec une vérification du nombre de côtés détectés et le nombre de côtés attendus (stipulé dans le dictionnaire), nous pouvons vérifier que nous avons bien reconnu le panneau. Il s'agit ensuite de récupérer les sommets du polygone détecté et approximé. Enfin, il suffit de calculer le centre avec des formules de géométrie classique en fonction du polygone à traiter. Le diagramme d'activité ci-dessous (3.3) modélise ce processus.

1

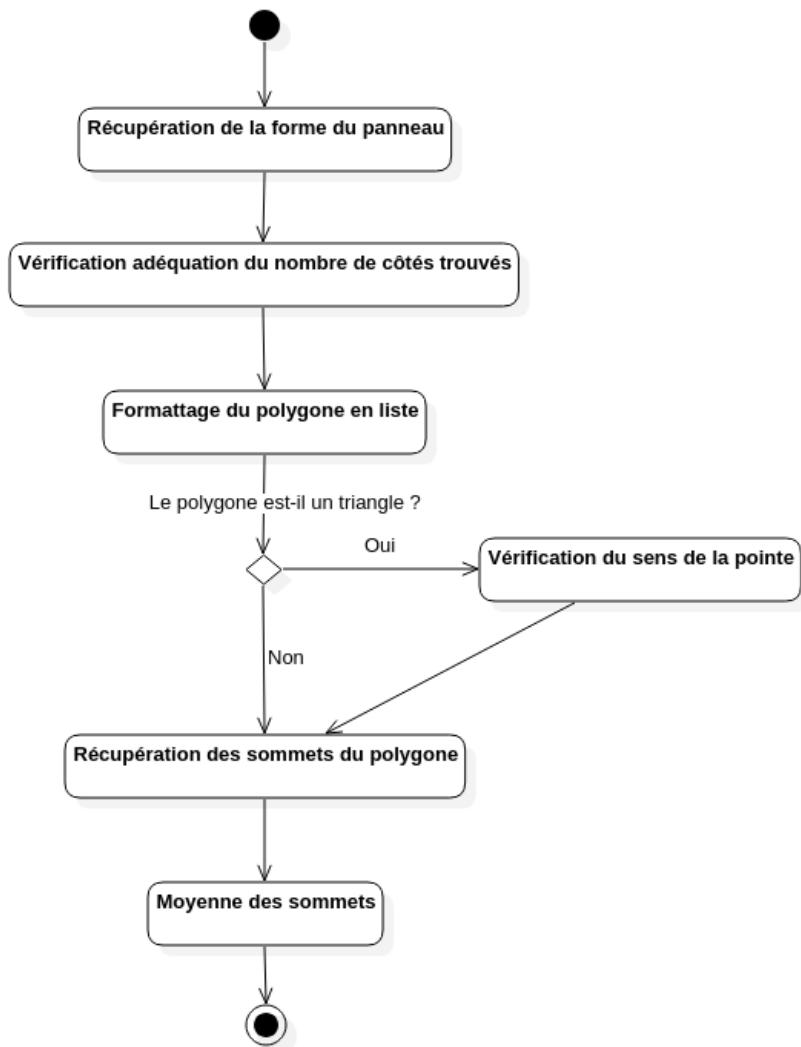


FIGURE 3.3 : Diagramme d'activité modélisant le calcul de centre de panneau

Nous avons ensuite procédé à la recherche de la **hauteur du panneau**. Nous souhaitions récupérer la valeur en pixels de la distance entre le haut et le bas du panneau. Pour cela, nous avons implémenté le modèle décrit dans le diagramme d'activité représenté sur la figure 3.6. La difficulté de cet algorithme a résidé dans le fait que les panneaux ont des formes géométriques imbriquées. Par exemple les panneaux de limitations de vitesse ont un liseré rouge sur le pourtour, assez large. Le contour intérieur est alors souvent détecté : il est souvent plus marqué et laisse moins de place aux ambiguïtés. Pour être sûr de prendre le contour extérieur du panneau et donc d'avoir sa hauteur réelle, nous effectuons le protocole suivant (voir figures 3.4 & 3.5) :

¹Canny, John. (1986). A Computational Approach To Edge Detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on. PAMI-8. 679 - 698. 10.1109/TPAMI.1986.4767851.

3. Analyse fonctionnelle - solution proposée

1. Récupérer les sommets du contour détecté ;
2. Regarder les couleurs à côté de ces points, du côté intérieur du contour (vers le centre) ;
3. Si les couleurs récupérées sont significativement rouge (dans la majorité des cas), alors c'est bien le contour extérieur qui a été détecté. On procède alors à un calcul de hauteur via les sommets déjà récupérés. Il s'agit du cas 1, à droite, sur la figure 3.4 ;
4. Si les couleurs ne sont pas significatives, cela signifie que c'est le contour intérieur qui a été détecté, comme c'est le cas sur la figure 3.4, à gauche ;
5. Dans ces cas là, on calcule l'intersection entre les droites créées par les sommets et passant par le centre, avec le contour détecté par filtre de Canny le plus proche du contour intérieur. Ceci est exemplifié sur la figure 3.5 ;
6. On considère alors que ces points d'intersection, en orange sur la figure 3.5 sont situés sur le contour extérieur du panneau, et on calcule la hauteur par rapport à ces points.

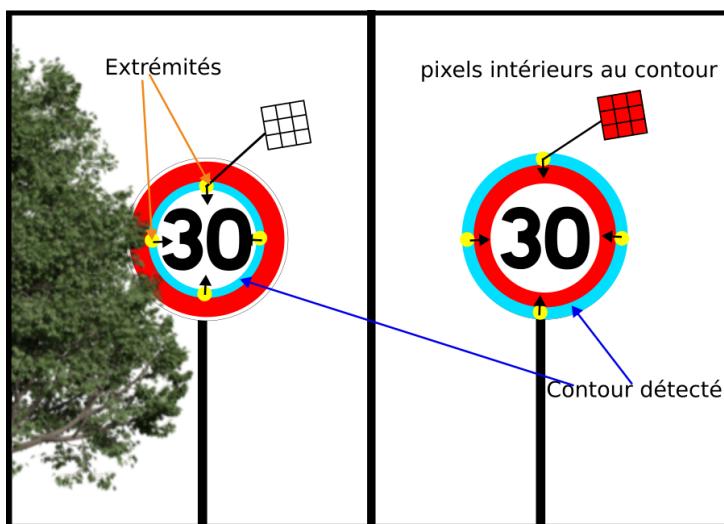


FIGURE 3.4 : Schémas des cas possibles lors du calcul de la hauteur

Orientation du panneau

En utilisant des outils similaires à ceux employés pour calculer le centre des panneaux, nous déterminons également l'**orientation relative** du panneau dans la photo recadrée. C'est-à-dire un angle d'orientation du panneau relative à l'imagette, compris entre -90° pour un panneau tourné vers la gauche et 90° pour un panneau tourné vers la droite, et 0° pour un panneau qui est en face. Cette orientation nous permet ensuite de déterminer l'**orientation réelle** du panneau. On combine le gisement et cette orientation relative pour obtenir la géolocalisation des panneaux et réaliser les intersections qui permettent de dédoublonner les panneaux. Pour ce faire, nous appliquons une chaîne de traitement similaire à celle utilisée pour calculer les centres (voir figure 3.3), en nous arrêtant à l'avant-dernière étape, la "Récupération des sommets du polygone". À partir de cette étape, nous utilisons les différents points repérés pour estimer la déformation et donc l'orientation du panneau, par calcul géométrique d'angles et de distance (voir figure 3.7). La valeur d'orientation obtenue est précise à 10° près mais son signe n'est pas précis et variable. Une autre méthode supplémentaire et complémentaire pourrait être envisagée pour déterminer la valeur plus précisément et le signe correctement. Elle a été évoquée à la fin du projet mais que très peu mise en oeuvre.

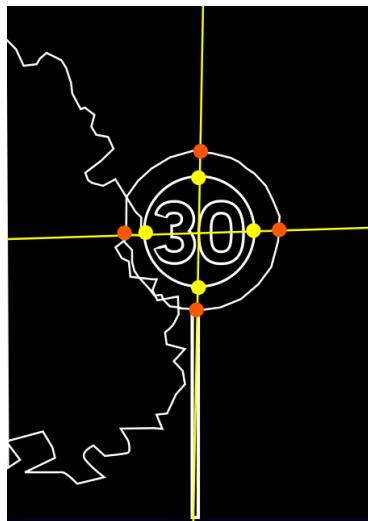


FIGURE 3.5 : Schéma montrant la récupération des sommets lors d'une détection de contour intérieur

3.2.2 Création d'une base de données intermédiaire

Comme indiqué précédemment, il nous a fallu retrouver et lier chaque imagette à son image originale d'où elle est tirée. Nous avons dû, pour chaque imagette, récupérer la géolocalisation située dans les tags EXIF, effectuer une requête spatiale sur l'API Panoramax, lire l'ID du résultat, et stocker à la fois les informations de l'imagette, de la photo dont l'imagette provient, ainsi que les informations de la séquence de la photo.

Modélisation d'une BDD de traitement

Nous avons traité une quantité importante de données (plus de 120 000 imagettes), il était donc nécessaire de modéliser une BDD différente de la base de données finale BD PANNEAU (modélisation en figure 3.8).

On retrouve 4 tables, dont la table de panneau finale :

- Cropped_sign : Table des imagettes, avec une clé étrangère indiquant de quelle photo l'imagette provient (`picture_id`) et l'id du panneau correspondant (`sign_id`, de valeur nulle à la création de la base de données). Le but de l'algorithme est donc de remplir cet id ;
- Picture : Table des photos originales, où sont stockées des informations nécessaires pour l'algorithme de détection des panneaux (hauteur, largeur, field of view / fov...), ainsi qu'une clé secondaire pointant vers la séquence dont elle provient ;
- Collection : Table des séquences, où la date de la séquence est indiquée. De par la construction du site Panoramax et de l'utilisation faite de la date de la photo, on peut considérer que la date de la séquence est équivalente à celle des photos.
- Sign : Table des panneaux détectés. Initialement, cette table est vide et n'est remplie qu'à la fin de la chaîne de traitement. Elle comporte des informations sur son orientation, sa taille, le type de panneau (en différenciant code et valeur) et la précision du calcul ;

Remplissage progressif de la BDD

Bien que mentionnée précédemment, nous n'avons pas explicité la manière dont nous avons rempli la base de données. Le remplissage s'est effectué en 3 parties, chacune reposant sur des algorithmes différents, expliqués dans les différentes sections de ce chapitre. Les parties sont les suivantes :

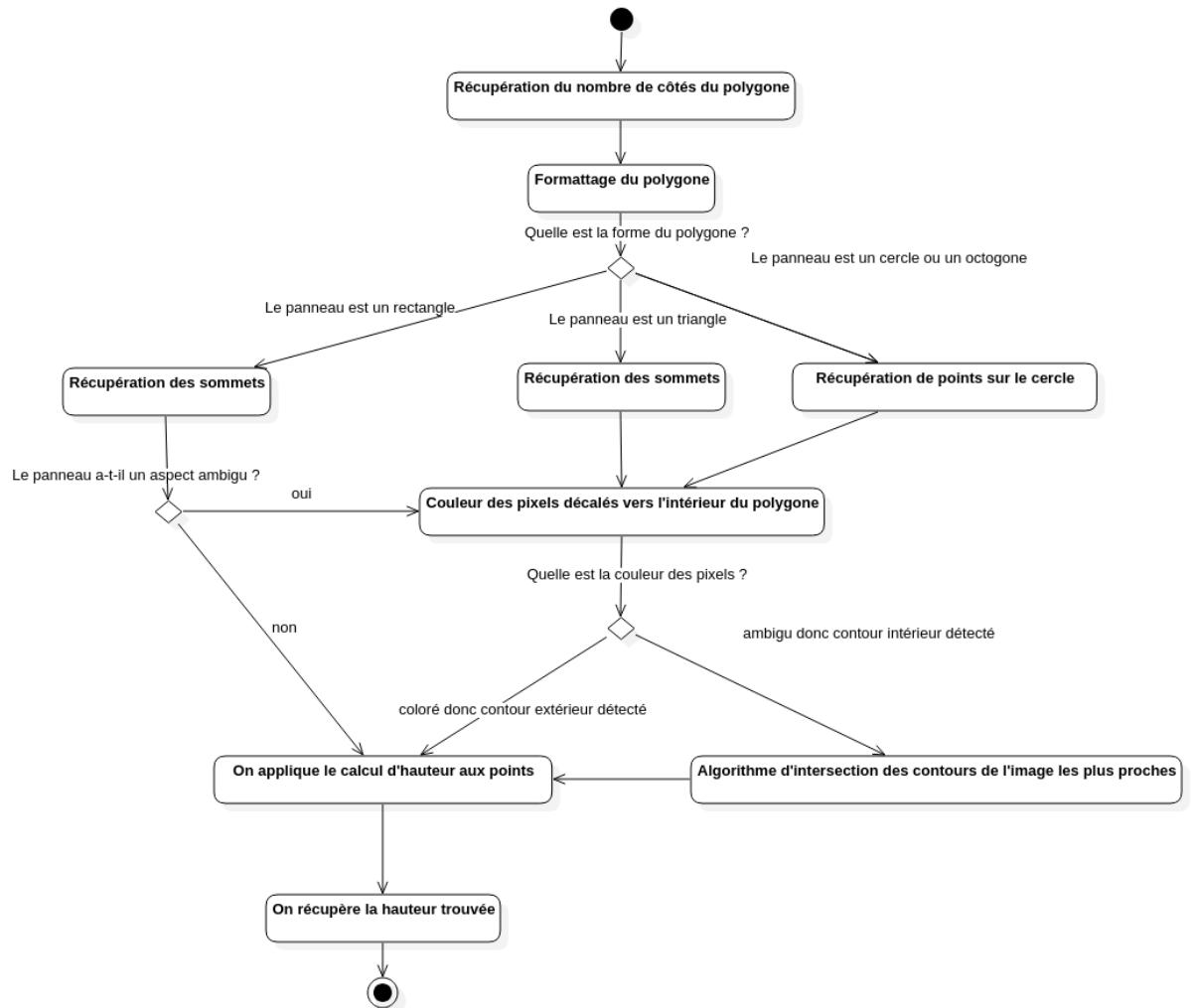


FIGURE 3.6 : Diagramme d'activité modélisant le calcul de hauteur de panneau

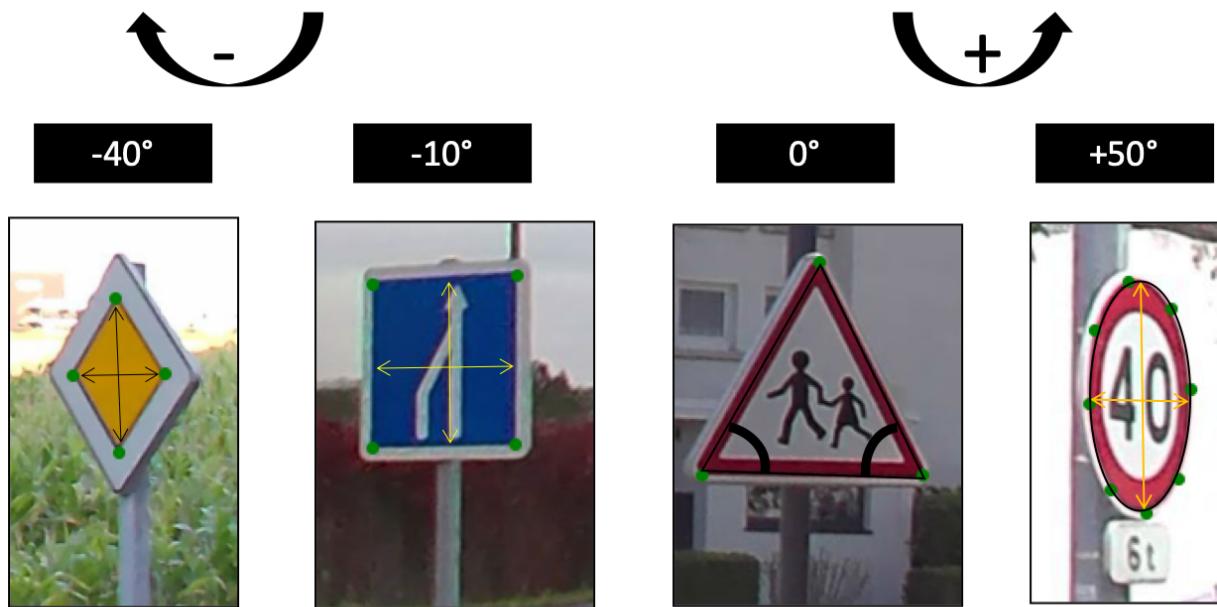
- **1^{ère} partie :** Remplissage de la base de données avec les informations primaires sur les imagettes, notamment via les métadonnées des imagettes et des appels à l'API *Search* ;
- **2^{ème} partie :** Remplissage de la base de données avec les informations calculées sur les imagettes : centre et hauteur du panneau, orientation du panneau ;
- **3^{ème} partie :** Détermination des panneaux et remplissage de la BD PANNEAU (explication dans la partie suivante).

3.2.3 Intersector les panneaux - Utilisation de méthodes inspirées de la photogrammétrie

Mesure du gisement

L'une des informations disponibles via l'API Panoramax concernant les photos est l'orientation de celles-ci. Si elle est fournie par l'utilisateur, elle est affichée telle quelle, sinon elle est recalculée. Elle indique l'angle entre l'axe de prise de vue de la photo et le nord.

La méthode à laquelle nous avons pensé était donc pour une photo, de répertorier tous les panneaux apparaissant dessus, et de tracer une droite partant du point de la photo dans la direction du panneau.



Imagette extraite de la détection de panneaux, après calcul de l'orientation

FIGURE 3.7 : Exemple de calculs d'orientation relative des panneaux dans les imagettes

Pour cela, on considère que toute la ligne supérieure de la photo est le zénith - point à la verticale de la prise de vue, en direction du ciel - et que toute la ligne inférieure est le nadir - point à l'opposé du zénith (visible fig 3.10). On peut noter qu'il est impossible de faire de telles approximations avec des photos planes, dont la géométrie comporte trop de paramètres inconnus, les rendant donc impossibles à exploiter.

À partir de ces axiomes, on peut considérer que la position horizontale du panneau correspond à son angle horizontal dans la sphère, et donc en déduire le gisement du panneau par rapport à la prise de vue (voir figure 3.9).

On va donc calculer la distance horizontale entre le centre de l'image et le centre du panneau, et reporter cette valeur en angle de cette façon :

$$\frac{x_{\text{pano}} \times 360}{x_{\text{tot}}} - 180$$

Où x_{pano} est la distance algébrique du centre de l'image au centre du panneau (en pixels), et x_{tot} est la largeur totale de l'image, en pixels, tel que montré dans la figure 3.10.

Mesure de la taille

Connaître la taille du panneau est très important pour savoir à quelle distance de la prise de vue il se trouve. En effet, si on connaît la taille s du panneau, et sa hauteur dz dans la photo, alors la distance d vérifie : $\sin\left(\frac{dz}{h} \times 180^\circ\right) \times d = s$ (voir figure 3.11)

On choisit de mesurer la taille du panneau verticalement, qui est invariant dans la projection dans une photo par rotation d'axe vertical.

On suppose que $dz \ll h$ donc $\sin\left(\frac{dz}{h} \times 180^\circ\right) \sim \frac{dz}{h} \times 180^\circ$.

On pose alors le facteur d'échelle (*size dist factor*) $sdf = \frac{h}{dz \times 180^\circ}$, vérifiant

$$d = sdf \times s$$

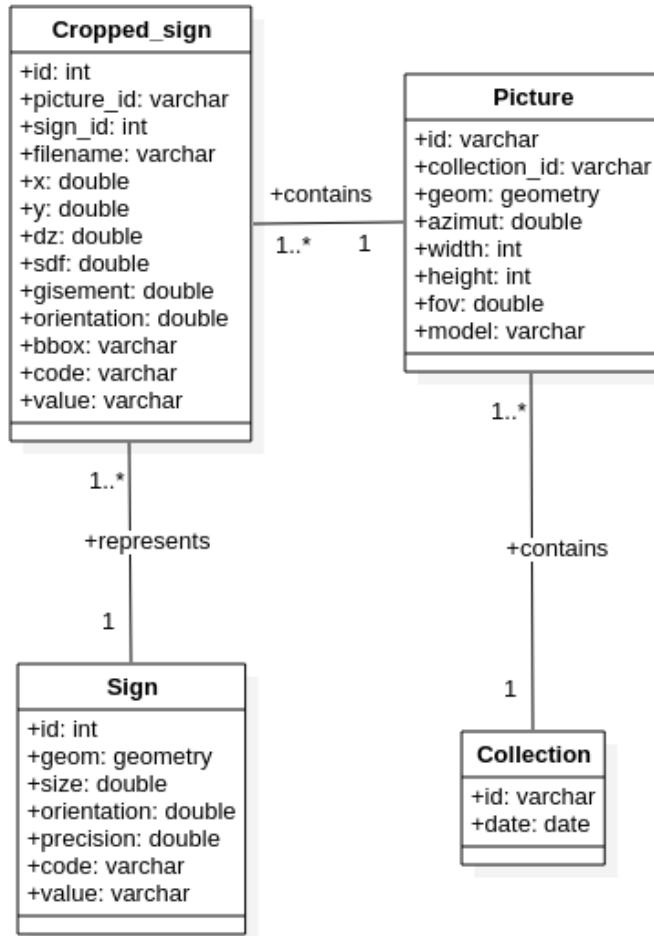


FIGURE 3.8 : Diagramme de classe représentant le modèle de données de la BD PANNEAU

Appareiller les détections de panneaux

Nous avons donc plusieurs détections de panneaux, caractérisées par :

- Un code et une valeur (ex : *limitation de vitesse à 30 km/h* : B14, 30)
- Un point de prise de vue, un gisement et un facteur d'échelle
- Une orientation

On cherche ici à regrouper les détections entre elles lorsqu'elles correspondent au même panneau physique. Chaque détection de panneau est une mesure, et non une valeur théorique exacte. On doit donc définir des critères de proximité et de compatibilité sur les détections (voir figure 3.12)

Notons f_{compat} la fonction qui à un ensemble de détections Ω associe

$$\begin{cases} 1 & \text{si les détections sont compatibles} \\ 0 & \text{sinon} \end{cases}$$

Soit la matrice des compatibilités $M = (m_{i,j} \in \{0, 1\}, (i, j) \in \llbracket 1..n \rrbracket^2)$, matrice carrée booléenne de taille n définie par $\forall (i, j) \in \llbracket 1..n \rrbracket^2, m_{i,j} = f_{compat}(\{p_i, p_j\})$. Par définition, cette matrice est symétrique, et sa trace vaut $n : M^T = M ; Tr(M) = n$

On cherche à clusteriser la matrice, en extrayant les plus grands groupes de détections compatibles entre elles. La matrice des compatibilités correspond à une clusterisation compatible si et seulement si

3. Analyse fonctionnelle - solution proposée

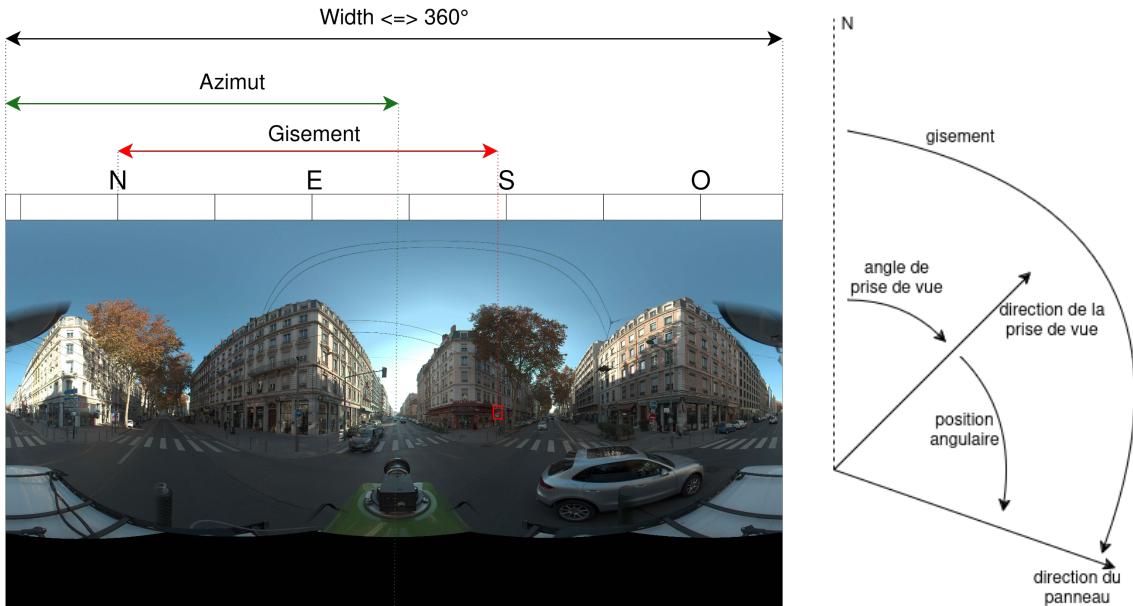


FIGURE 3.9 : Lecture du gisement dans une prise de vue

elle peut être réorganisée par ré-indexation des panneaux (i.e. en appliquant une même permutation des lignes et des colonnes) en une matrice diagonale par blocs. Cela peut se traduire par l'égalité des lignes au sein d'une clusterisation. Voir figure 3.14.

La matrice des compatibilité n'est pas nécessairement clusterisée compatible. On cherche donc une matrice M' clusterisée compatible avec un minimum de clusters. Afin de garantir la compatibilité, M' doit vérifier $\forall(i, j), m_{i,j} \geq m'_{i,j}$. On utilisera la notation suivante pour exprimer les lignes (idem pour les colonnes) $m_{i,:} = (m_{i,j}, j \in [1, n])$

On construit la matrice des compatibilité semblables $S_{i,j} = g(m_{i,:}, m_{j,:}) * m_{i,j}$ avec g une distance sur les lignes de M . On prends ici $g : A \in \{0, 1\}^n, B \in \{0, 1\}^n \mapsto \frac{\sum a_i \& b_i}{\sum a_i | b_i} \in [0, 1]$ avec $\&$, $|$ les opérateurs binaires ET et OU.

Nous abuserons de la notation pour écrire $S(M)$ la matrice des compatibilités semblables issue de M . S est donc la fonction qui à une matrice de compatibilité associe sa matrice des compatibilités semblables.

$S(M)$ vérifie bien par définition la propriété $\forall(i, j), m_{i,j} \geq m'_{i,j}$.

On peut donc construire un algorithme de clusterisation de matrice de compatibilités : voir figure 3.15.

Ainsi, on a regroupé les détections de panneaux en panneaux compatibles.

Générer la BD PANNEAU en prenant en compte l'existant

La BD PANNEAU est publique. Il faut donc veiller à ne pas casser ses utilisations en la mettant à jour :

- Les id (numéro d'identification) des panneaux ne doivent pas changer
- Les panneaux existants ne doivent pas trop bouger

On va donc chercher à évaluer les panneaux détectés pour savoir s'ils sont suffisamment fiables pour être intégrés dans la BD PANNEAU, puis à les appareiller avec les panneaux existants pour assurer la stabilité des id.

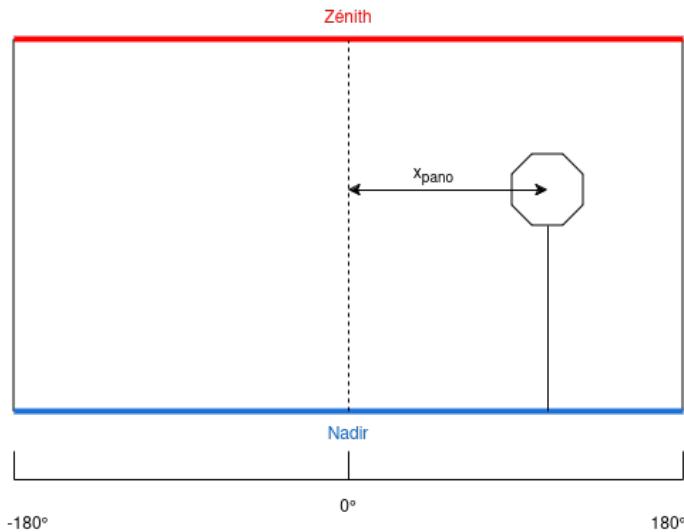


FIGURE 3.10 : Schéma du calcul largeur/angle

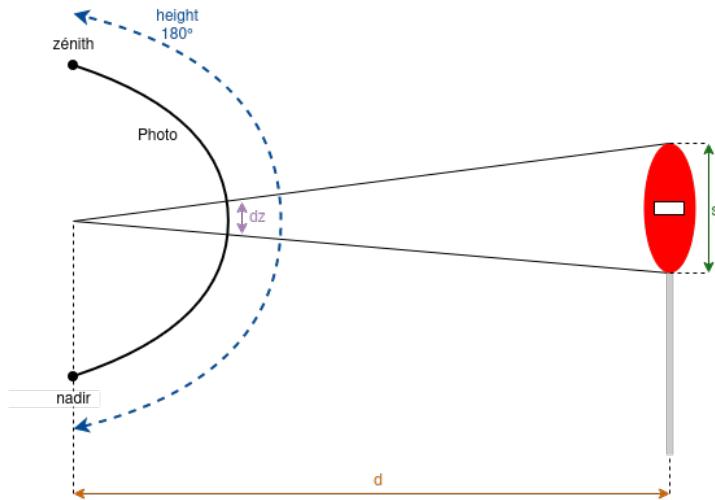


FIGURE 3.11 : Calcul du facteur d'échelle

Un nouveau panneau est fiable s'il a suffisamment de détections associées (par défaut : 5), dont un certain nombre vu de face (par défaut : 3).

On peut alors appareiller les panneaux détectés fiables aux panneaux existants en se basant sur les critères de compatibilité exprimés précédemment (voir figure 3.12).

Les nouveaux panneaux fiables seront ajoutés et les panneaux existants seront mis à jour.

3.3 Évaluation de la qualité

Panoramax étant une plateforme collaborative, la qualité des données n'est pas vérifiée selon des critères lourds, ce qui permet à un maximum de personnes de contribuer. Cependant, cela peut poser problème lors de l'utilisation de ces données. Il est toujours intéressant de s'intéresser à la qualité des données et algorithmes produits, mais cela l'est d'autant plus lors de l'utilisation de données collaboratives.

- Si plusieurs détections ont un **code** ou une **valeur** incompatible, ces dernières sont incompatibles entre elles. À noter qu'un panneau peut être détecté de dos, noté par le code A00, mais sa forme ne sera pas traitée ; alors que dans une version ultérieure de la classification, les formes seront discriminées, ce qui affinera les résultats.

Par exemple, un panneau de limitation à 30 km/h pourra être détecté ainsi :

- Panneau vu de face : code B14, valeur 30
- Panneau vu de dos : code A00, valeur None

- Le deuxième critère est plus complexe. Utilisons le schéma 3.13 afin de l'expliquer. On considère les détections 1, 2 et 3 représentées par des croix. Ces détections sont caractérisées par leur **localisation**. Elles ont également chacune un **gisement**, l'angle représenté entre le Nord et l'angle du panneau.

Sur le schéma, pour chaque point, le Nord est représenté par une flèche verticale dirigée vers le haut, et l'angle du panneau par une droite en pointillé. Le gisement est donc l'angle entre ces deux figurés. Enfin, elles sont caractérisées par un **facteur d'échelle**, rapport entre la taille du panneau dans la détection et la distance à l'appareil. La taille du panneau est représentée (relativement à la position de l'appareil) par les ronds.

Si des détections ont des points, gisements et facteurs d'échelle incompatibles, elles sont incompatibles. Les points, gisements et facteurs d'échelle sont compatibles s'il existe un panneau (position, taille) suffisamment proche (critère de distance 2D par défaut : 5m, critère d'écart de taille : 0.2m) ;

- Si les détections ont des **orientations** incompatibles, elles sont incompatibles. Les orientations sont compatibles si elles sont suffisamment proches (à 360° près). On regarde la distance maximale entre chaque angle, et on la compare à un seuil (par défaut : 45°) ;
- Sinon, les détections sont dites **compatibles**.

FIGURE 3.12 : Critères de compatibilités

3.3.1 Critères de qualités minimum sur les données

Bien que le projet ne se concentre pas sur l'évaluation de qualité de la donnée, il a été nécessaire de formuler des critères minimums pour pouvoir utiliser les données. Ces critères sont assez simples, mais permettent de faire un premier tri sur les données.

Le critère le plus important est un des plus simples à mettre en œuvre. Il concerne le type de photos utilisé. En effet, nous n'utilisons dans l'algorithme de localisation des panneaux que des images 360°. L'utilisation restreinte de ces images nous permet notamment de confirmer l'hypothèse sur le nadir et le zénith évoquée dans la partie 3.2.3, afin de ne pas avoir à calculer des paramètres photogrammétriques intrinsèques aux photos 180°.

D'autres critères ont été choisis pour définir si on peut traiter une imagette pour le calcul du centre et de la hauteur. Premièrement, si le **nombre de côtés du plus grand polygone fermé détecté dans l'imagette est supérieur à vingt**, l'image n'est pas traitable. Ce choix s'explique par le fait que les panneaux ont normalement huit côtés maximum (cas des panneaux stop), donc vingt côtés détectés correspondent nécessairement à une erreur. Il ne faut pas faire un choix trop restrictif à ce stade (par exemple enlever les imagettes dont le nombre de côtés est supérieur à neuf) car les algorithmes développés n'utilisent pas de technologies assez avancées (comme de l'apprentissage profond pour détecter les formes des panneaux) pour s'assurer que nos résultats sont toujours bons et conformes à ce que l'on attend. Le deuxième critère correspond à l'aire de la forme du panneau

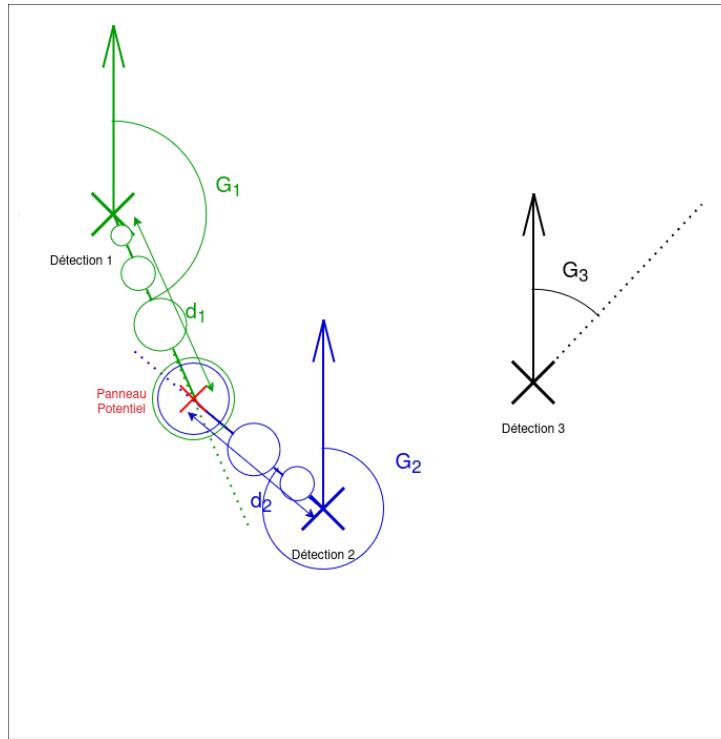


FIGURE 3.13 : Explication second critère

détecté. Les imagettes étant centrées sur les panneaux, il n'est pas possible d'avoir une aire trop petite par rapport à la taille de l'image. On calcule donc l'aire détectée et si elle est **inférieure à 5% de la taille de l'image**, l'aire détectée n'est définitivement pas celle du panneau et on ne peut pas calculer la hauteur et le centre du panneau. Dans la majorité des cas, c'est ce dernier critère qui est déterminant, car il arrive assez fréquemment que la forme détectée ne soit pas le panneau. La figure 3.16 montre des exemples de panneaux dont la forme détectée est trop petite.

3.3.2 Précision de l'algorithme de détection de centre et de hauteur

Dans notre chaîne de traitement, l'évaluation du centre et de la hauteur du panneau est importante pour la localisation des panneaux. De ce fait, il est important d'évaluer la précision de cet algorithme, car la précision finale des panneaux va en dépendre. Pour évaluer la précision sur un maximum de panneaux différents, nous avons décidé de partir de la base de données de traitement pour extraire 10 imagettes de chaque type de panneau. Pour cela, nous parcourons les différents codes du dictionnaire mentionné au début de la partie 3.2.1, et nous faisons des requêtes SQL à la base de données pour extraire 10 imagettes de chaque code. Les tests sont effectués et un premier tri est fait entre les cas traités et ceux qui n'ont pas pu l'être (car le centre trouvé n'est pas le bon ou qu'une exception s'est produite). Le centre et la hauteur sont dessinés sur les imagettes et enregistrés en local. Sur les images ayant été testées, un contrôle visuel est fait pour différencier les images avec des bonnes valeurs des autres. Les "mauvaises" images sont ensuite discriminées en plusieurs catégories : centre bon, hauteur bonne, seulement centre bien détecté, mauvaises valeurs et résultats aberrants (correspondant souvent à un centre mal détecté et pas de hauteur calculée). Des images correspondantes à ces différents cas sont représentées sur la figure 3.17

Cela nous donne donc un résultat visuel, mais pas statistique. Les tests ont été réalisés sur 1202 imagettes, et nous avons donc pu réaliser un tableau représentant les résultats selon les différents types de panneau (les types de panneau sont représentés sur la figure 3.2). La figure 3.18 représente ce tableau.

La première analyse de ces résultats montre que seulement un panneau sur deux a des bonnes

3. Analyse fonctionnelle - solution proposée

Clusterisations compatibles	Clusterisation non compatible
$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$

FIGURE 3.14 : Exemple de clusterisations compatibles et non compatibles

Input :

- $p_i, i \in \llbracket 1, n \rrbracket$: détections de panneaux

Paramètres :

- n_itter : nombre d'itérations

Initialisation : Construire M_0 la matrice des compatibilités issue des détections de panneaux

$$\text{Étape } k \in \llbracket 1, n_itter \rrbracket : M_k = \left(S(M_{k-1}) \geq \frac{k}{n_itter} \right)$$

Fin : $M' = M_{n_itter}$
 M' est ainsi clusterisée.

FIGURE 3.15 : Algorithme de clusterisation de matrice de compatibilité

valeurs et un panneau sur quatre n'a pas de centre trouvé (ligne `pb_none`). Sur ce dernier résultat, les types de panneaux ayant le plus de problèmes sont les panneaux de type 5, i.e. les panneaux carrés. Cela s'explique par la complexité de la détection du centre des panneaux carrés comparé aux panneaux ronds ou triangulaires (il est plus compliqué d'obtenir des extrémum avec un carré qu'avec un rond ou un triangle). La majorité des panneaux sont de type 0 (triangle), 4 (rond) ou 5 (carré). Les panneaux ronds sont ceux avec les meilleurs résultats (3/4 des panneaux sont bien détectés), tout comme les panneaux triangulaires (2/3 des panneaux). Comme mentionné précédemment, les panneaux carrés sont ceux avec le plus de problèmes (seulement 1/5 des panneaux sont bien détectés).

Pour pallier ces problèmes, il serait possible d'améliorer les algorithmes ou à minima de prendre des valeurs arbitraires, notamment en cas d'exception. Par exemple, le centre du panneau peut être le centre de l'image, et la hauteur de celui-ci peut correspondre à la moitié de la hauteur de l'image originale. En faisant cela, il reste important de garder en mémoire (i.e. via un champ de la base de données) si les résultats ont été calculés ou non (car cela influe forcément sur la qualité de la détection ensuite).

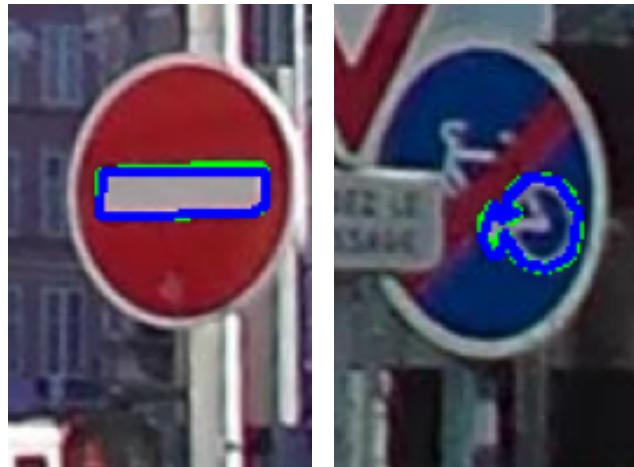


FIGURE 3.16 : Imagettes n'ayant pu être traitées car les formes détectées sont trop petites

3.3.3 Précision de l'algorithme de localisation des panneaux

Tout comme pour la hauteur et le centre, nous avons pu évaluer la précision de l'algorithme produit pour la localisation des panneaux. Pour cela nous avons mis au point un système de score de confiance, basé sur 3 éléments :

- Plus un panneau est aperçu sur des détections différentes, plus il est fiable
- Plus un panneau est vu sous une grande variété d'angles, plus la position estimée est fiable
- Lorsqu'on calcule la localisation d'un panneau, on estime aussi sa hauteur réelle (en m). Si la hauteur estimée est en dessous de 10cm ou au dessus de 10m, elle est considérée aberrante.

Ce score est donc le produit de trois coefficients, tous compris entre 0 et 1, ce qui donne un score de confiance final compris dans cet intervalle.

- Coefficient 1 :

$$\frac{\arctan(0.45 * (x - 5))}{\pi} + 0.5$$

, où x est le nombre de détections du panneau. Cela permet d'avoir une valeur de 0.5 pour 5 détection, qui augmente rapidement jusqu'à 10 détections, puis qui croît plus lentement jusqu'à 15 détections avant de stagner.

- Coefficient 2 : En considérant tous les angles sous lesquels est aperçu le panneau dont il est question, on a :

$$1 - \left((\text{mean}(\cos(\theta))^2 + \text{mean}(\sin(\theta))^2 \right)^2$$

Ce qui donnera une valeur proche de 1 si les angles θ sont dispersés, et proche de 1 s'ils sont similaires.

- Coefficient 3 : 0 si la valeur de taille calculée est aberrante, 1 sinon.

On peut donc noter que pour un panneau qui n'est détecté qu'une fois, le coefficient n°2 sera nul car il ne sera vu que d'un seul angle. Cela reste cohérent avec le fait qu'on ne peut simplement pas estimer la position d'un panneau avec une seule détection.

3. Analyse fonctionnelle - solution proposée



FIGURE 3.17 : En haut, de gauche à droite : bonnes valeurs, centre ok, hauteur ok ; en bas, de gauche à droite : seulement le centre, mauvaises valeurs et résultats aberrants

Type	0	1	2	3	4	5	6	7	8	9	Total	Proportion
height_ok	1	0	0	0	14	30	1	0	0	0	46	4%
center_ok	3	0	1	0	15	12	0	0	0	0	31	3%
bad_values	2	0	1	1	47	20	0	0	0	0	71	6%
pb exception	0	0	0	0	1	3	0	0	0	0	4	0%
pb none	67	0	0	5	0	168	30	20	0	0	290	24%
pb non-treatable	2	0	1	0	36	3	0	0	0	0	42	3%
only_center	0	0	0	0	2	72	0	0	0	0	74	6%
nothing_good	0	0	0	0	5	12	2	0	0	0	19	2%
good_signs	182	0	7	8	339	85	4	0	0	0	625	52%
Total	257	0	10	14	459	405	37	20	0	0	1202	100%

FIGURE 3.18 : Résultats sur le jeu test

4. Étude technique

4.1 Langages, techniques et outils utilisés

Nous nous sommes accordés avec les commanditaires dès le début du projet sur les différents outils qui seraient utilisés. Une des contraintes principales est l'*open source*. En effet, puisque nous travaillons en collaboration avec OpenStreetMap (OSM), nous nous devons de respecter les méthodologies du collaboratif. Étant donné que nous avons traité des images et effectué plus généralement du calcul scientifique (photogrammétrie), le langage python s'est imposé dès le début comme langage principal. De plus, il existe de multiples bibliothèques python *open source*. Nous avons notamment utilisé la bibliothèque **OpenCV**, une bibliothèque de traitement d'images.

Ensuite, nous avons utilisé une base de données **PostgreSQL** classique pour stocker les données, ainsi que l'extension **PostGIS** pour gérer le volet géographique de nos données. Le langage de requête est donc, *a fortiori*, le **Structured Query Language (SQL)**. Nous avons, par ailleurs, mis en place un poste en tant que serveur afin de faciliter l'accès à une base de données commune par l'équipe.

Nous avons travaillé en utilisant les méthodes AGILEs. Ainsi, nous avons convenu avec les commanditaires d'une réunion hebdomadaire pour discuter de nos avancées, problèmes et questions. De plus, les commanditaires nous ont proposé de mettre en place un canal de discussion en direct pour éviter d'envoyer des mails, qui auraient pu ralentir les échanges.

Concernant les outils utilisés, nous avons utilisé **GitHub**, dont l'utilisation sera détaillée dans la partie suivante. Nous avons également utilisé **pgAdmin** pour faciliter l'interaction avec notre base de données.

4.2 Gestion de versions, intégration et déploiement continu

4.2.1 Git

Comme spécifié précédemment, nous avons utilisé GitHub. La technologie Git est très souvent utilisée dans les projets de développement, car elle permet de partager du code entre plusieurs utilisateurs, mais sans forcément que les modifications de l'un impactent directement celles de l'autre, grâce au système de branche. Nous avons défini la branche *main*, à partir de laquelle nous avons créé de nouvelles branches pour développer des fonctionnalités, avant de les joindre de nouveau sur la branche *dev* (branche de développement) lorsque celles-ci étaient prêtes.

De plus, GitHub permet d'effectuer des *GitHub Actions*. Ce sont des processus que l'on définit, qui se déclenchent lorsque l'on effectue une certaine action sur le dépôt Git associé. Nous nous en sommes servis pour tester notre code à chaque fois que nous le poussions sur ce dernier.

4.2.2 Docker

Afin d'être raccord entre nous sur les données sur lesquelles nous travaillons, nous avons décidé de faire tourner notre base de données sur un conteneur Docker, sur un ordinateur tiers. Cela nous a permis de mieux contrôler l'environnement, et de répliquer cette base de données pour nos *github actions*. Nous avons donc composé un fichier *docker-compose* afin de reproduire ce conteneur facilement.

4.3 Architecture technique et physique de la solution

La conception de la solution repose principalement sur une chaîne de traitement, ce qui rend difficile l'application d'une architecture en couches ou en tiers classique. Cette particularité découle de notre objectif de développer des processus et des entités exclusivement au niveau du *backend*. Néanmoins, l'organisation des différentes briques logicielles peut être mieux appréhendée à travers le diagramme de composants présenté ci-dessous (figure 4.1).

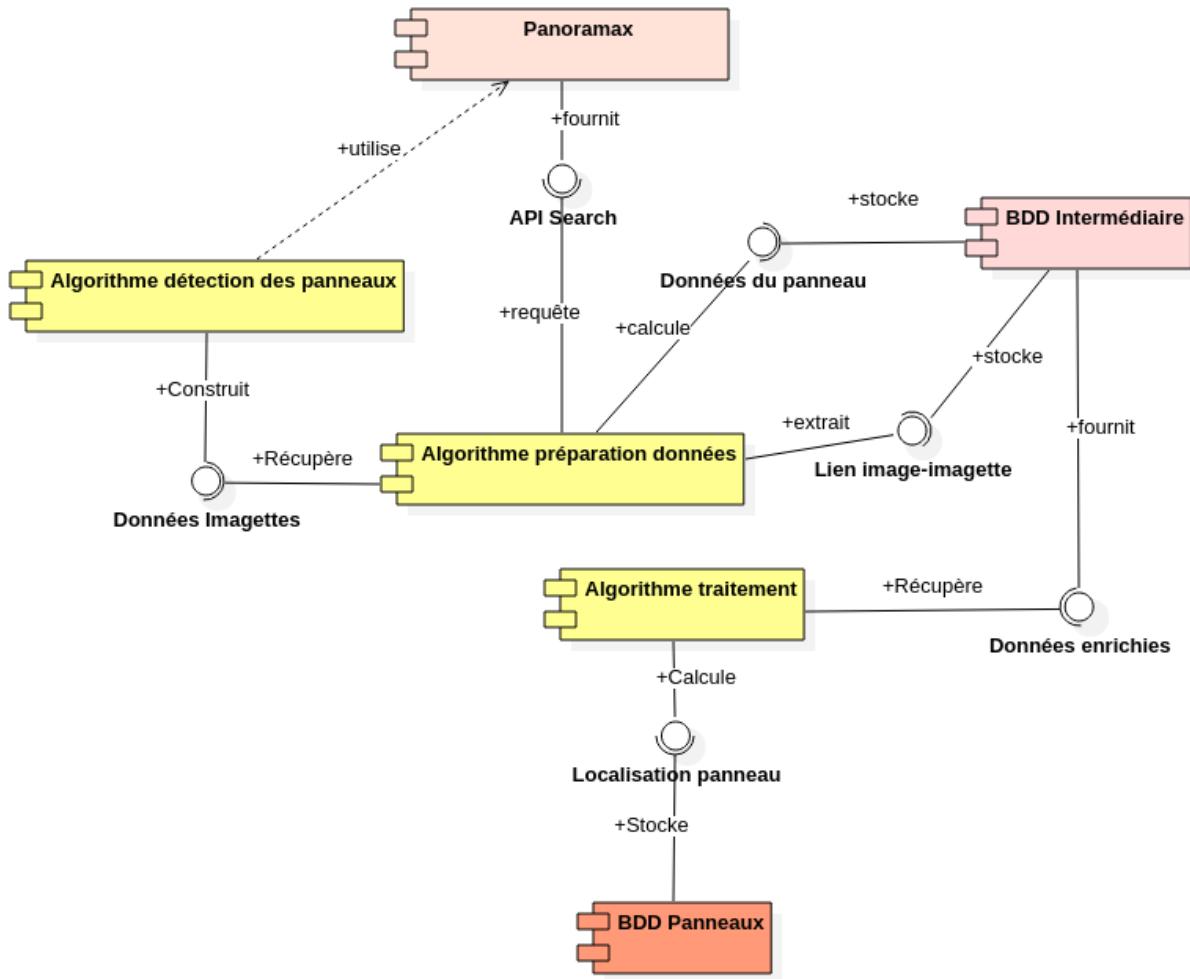


FIGURE 4.1 : Diagramme de composants

Comme on peut le voir, on considère six briques logicielles : deux qui sont des bases de données, trois qui sont des algorithmes, et enfin le site web de Panoramax. On peut retrouver la base de données intermédiaire qui stocke toutes les informations nécessaires au calcul de localisation des panneaux. Ces calculs permettent eux-mêmes de stocker la localisation dans la BD PANNEAU. On retrouve aussi l'algorithme de détection des panneaux basé sur les images de Panoramax que nous n'avons pas réalisé. En effet, nous avons simplement récupéré les imagettes déjà découpées mais avons jugé intéressant de l'intégrer au diagramme de composant puisqu'il complète bien le *backend* implémenté et son organisation générale.

Il est intéressant de noter que concrètement, le composant BDD intermédiaire et le composant BDD Panneaux sont les mêmes, à la différence près que pour le côté BDD Panneaux, la table sign représentée dans la figure 3.6 est complétée et représente le cœur de la base.

5. Réalisation et suivi du projet

5.1 Phasage du projet

Dès le début du projet, nous avons clairement défini nos deux objectifs principaux : dédoublonner les détections des panneaux et les géolocaliser. Pour structurer notre approche, nous avons établi un rétroplanning, comme illustré dans la figure 5.1. Ce plan nous a permis de segmenter nos objectifs généraux en tâches plus précises, facilitant ainsi la gestion de notre progression.

MARS			AVRIL		
1	Ven		1	Lun	Férié
2	Sam		2	Mar	RDV Panoramax- Point
3	Dim		3	Mer	
4	Lun		4	Jeu	RDV Gendarme - Cas d'usage
5	Mar		5	Ven	
6	Mer		6	Sam	
7	Jeu		7	Dim	
8	Ven		8	Lun	Vacances
9	Sam		9	Mar	
10	Dim		10	Mer	
11	Lun		11	Jeu	
12	Mar		12	Ven	
13	Mer		13	Sam	
14	Jeu		14	Dim	
15	Ven		15	Lun	RDV Panoramax- Point
16	Sam		16	Mar	
17	Dim		17	Mer	
18	Lun		18	Jeu	
19	Mar		19	Ven	
20	Mer		20	Sam	
21	Jeu	Présentation des projets	21	Dim	
22	Ven		22	Lun	RDV - Panoramax
23	Sam		23	Mar	V0 rapport
24	Dim		24	Mer	
25	Lun	RDV IGN- Précisions	25	Jeu	
26	Mar		26	Ven	Rendus Panoramax
27	Mer	RDV IGN - Cas d'usage	27	Sam	
28	Jeu	RDV - C. Quest	28	Dim	
29	Ven	RTT	29	Lun	RDV IGN - soutenance blanche
30	Sam		30	Mar	Soutenance
31	Dim				

FIGURE 5.1 : Rétroplanning effectué en début de projet

Nous avons aussi effectué un diagramme de Gantt, représenté sur la figure 5.2, afin de paralléliser les actions et d'estimer le temps nécessaire pour chaque objectif.

Bien sûr, ce diagramme de Gantt a évolué au cours du projet. Aussi, nous avons rapidement été amenés à redéfinir des sous-objectifs permettant, techniquement, d'avancer dans le projet. Par exemple, nous avons défini les sous-objectifs "calculer la hauteur des panneaux", "calculer le centre

5. Réalisation et suivi du projet

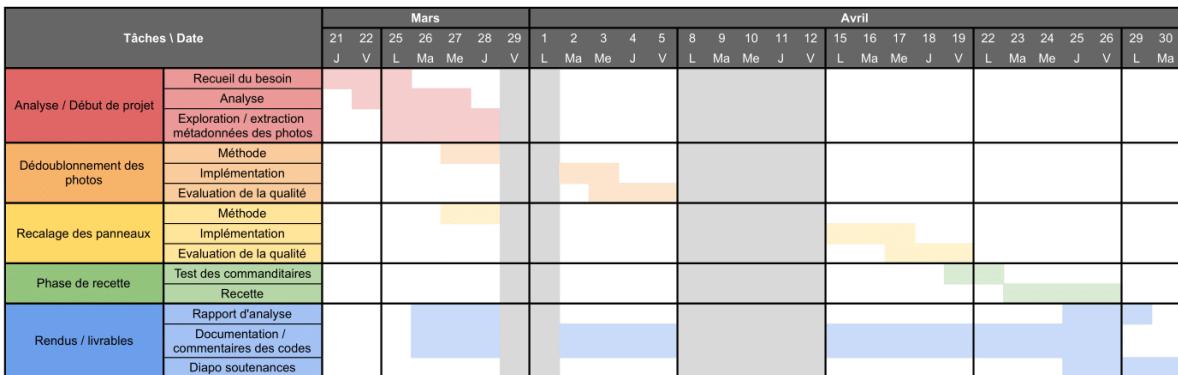


FIGURE 5.2 : Diagramme de Gantt effectué au début du projet

des panneaux", "extraire les informations des imagettes" et "remplir la base de données avec les informations actuellement disponibles". Toutes ces tâches ont été détaillées en interne de notre organisation via l'usage d'un Kanban effectué sur Trello.

Enfin, afin de conserver notre vigilance quant au bon déroulement du projet, nous avons élaboré une matrice des risques évolutive de semaine en semaine pour évaluer les "menaces" du projet au jour le jour. Les figures 5.3 et 5.4 sont les matrices des risques de la deuxième et de la dernière semaine. On peut y voir l'évolution des probabilités que les risques se concrétisent.

Risques	Mécompréhension avec les commanditaires	API Non Dispo	Résolution des images non exploitable	Orientation des panneaux impossible à exploiter	Ne pas avoir accès aux données en télétravail	Précision de la géoloc mauvaise
Probabilité	Faible	Moyenne	Moyenne	Moyenne	Moyenne	Moyenne
Impact	Léger	Important	Modéré	Léger	Léger	Modéré
Conséquences	Fausse route	Ralentissement des processus	Baisse de précision des résultats	Ne pas connaître la direction du panneau	Être moins efficace à distance	Baisse de précision des résultats
Solution préventive	Réunions hebdo	(Lancer les process pendant la nuit)	Déterminer le minimum suffisant	Faire des tests	Créer un pont de la BDD locale vers l'extérieur	Déterminer le minimum suffisant
Solution Corrective	Canal Discussion	Notifier les commanditaires en cas de problème	Jeter les données trop mauvaises		S'envoyer les fichier / les informations par upload	Jeter les données trop mauvaises
Évolution	-	-	-	-	-	-

FIGURE 5.3 : Matrice des risques de la deuxième semaine du projet

Les commanditaires nous ont rapidement proposé de nous rencontrer à Saint-Mandé dans les locaux de l'IGN pour que nous rencontrions l'équipe. Nous y sommes retournés une seconde fois la semaine suivante pour rencontrer et discuter avec des potentiels utilisateurs d'une BD PANNEAU.

Tout au long du projet, nous avons participé à des réunions hebdomadaires pour parler de notre avancement et de nos problèmes aux commanditaires, conformément aux méthodes AGILE.

Enfin, nous avons utilisé Matrix comme canal de discussion en direct avec les commanditaires, et Jitsi pour les réunions.

Pour résumer, nous avons suivi et géré notre projet en utilisant des méthodes et des façons de faire provenant des méthodes AGILE : réunions hebdomadaires, application des méthodes agiles, segmentation des objectifs généraux en tâches techniques. Pour appliquer ces méthodes, nous avons utilisé des outils : rétroplanning, diagramme de Gantt, matrice des risques, Kanban, réunions régulières avec les commanditaires et l'équipe de développement.

Risques	Mécompréhension avec les commanditaires	API Non Dispo	Résolution des images non exploitable	Orientation des panneaux impossible à exploiter	Ne pas avoir accès aux données en télétravail	Précision de la géoloc mauvaise	Nombre de données trop important à
Probabilité	Faible	Faible	Moyenne	Forte	Faible	Forte	Forte
Impact	Léger	Important	Modéré	Léger	Léger	Modéré	Important
Conséquences	Fausse route	Ralentissement des processus	Baisse de précision des résultats	Ne pas connaître la direction du panneau	Être moins efficace à distance	Baisse de précision des résultats	Moins de panneaux détectés
Solution préventive	Réunions hebdo	(Lancer les process pendant la nuit)	Déterminer le minimum suffisant	Faire des tests	Créer un pont de la BDD locale vers l'extérieur	Déterminer le minimum suffisant	Améliorer les détections de paramètres
Solution Corrective	Canal Discussion	Notifier les commanditaires en cas de problème	Jeter les données trop mauvaises		S'envoyer les fichiers / les informations par upload	Jeter les données trop mauvaises	
Évolution	-		-				
Etat	Faible	Léger	Forte	Moyenne	Léger	Très forte	

FIGURE 5.4 : Matrice des risques de la dernière semaine du projet

5.2 Travail réalisé

Voici l'état des lieux du travail réalisé au cours de ce projet :

- Une base de données reliant imagettes et images, enrichie de plusieurs données : la BD PAN-NEAU ;
- Les divers codes pythons commentés nous ayant permis d'arriver à ce résultat ;
- Des tests automatiques permettant l'intégration continue, nous prévenant dès lors que l'on provoquait un changement néfaste qui interférait avec les versions précédentes ;
- Une conceptualisation et une analyse logicielle, détaillée dans ce rapport.

5.3 Problèmes rencontrés

Des problèmes d'origines différentes ont été rencontrés à différents stades du projet. La plupart des problèmes ont déjà été évoqués dans les parties précédentes, mais nous les récapitulons ici :

- **Qualité des données** : La qualité des données a été un problème impactant la plupart des algorithmes que nous avons faits, à commencer par la détection du centre et de la hauteur qui n'est souvent pas bonne car la détection des contours n'est pas possible. Celui-ci étant important pour l'algorithme de détection des panneaux, la qualité des données impacte chaque résultat du projet. Lorsque l'on parle de qualité, on parle de qualité au niveau de : la résolution, la géolocalisation, le manque d'informations quantitatives concernant la qualité de ces données et d'informations sur des paramètres nécessaires pour la photogrammétrie classique.
- **Éléments du paysage** : Certains éléments du paysage (buisson, grille, voiture etc.) peuvent aussi empêcher la détection de panneau sur les images, bloquant la ligne de vue du panneau sur les photos. Cela n'est pas dû à la qualité des données en elle-même, mais empêche toutefois de détecter les panneaux.
- **Réception des imagettes** : Bien que ce problème ne soit pas technique, nous avons mis du temps à récupérer les imagettes ce qui a eu un impact sur le reste du projet (celui-ci ne durant que 4 semaines en tout).
- **Localisation des images** : Les images que nous avons utilisé se sont avérées être très mal positionnées malgré la technique RTK, qui est censée pouvoir être précise centimétriquement. Nos calculs sont donc forcément impactés par ces décalages, et perdent largement en précision.

5.4 Évolutions possibles

Comme évoqué dans la partie précédente, nous avons rencontré Matthieu Le Masson, responsable pilotage des projets à l'IGN. Il serait intéressé par une BD PANNEAU venant enrichir la BD TOPO®. Nous avons également dialogué avec Romain Veldeman, un gendarme qui nous a parlé de l'usage qu'il pourrait faire d'une telle base de données. En synthétisant ces deux échanges, on peut alors imaginer les évolutions qui découleront éventuellement de notre projet. Si l'on parvient à obtenir la localisation (au mètre près) de tous les panneaux d'une zone, il sera possible d'assigner des valeurs (de limitation de vitesse, de hauteur ou de poids maximums) à des tronçons de route, pour du calcul d'itinéraire. Au coeur de l'IGN, un travail étudiant pourrait porter sur l'exploitation et l'utilisation d'une telle base de données. Il nous a donc été fourni des idées, des ébauches de travail déjà réalisées et des réflexions à ce sujet, pour pouvoir amorcer la suite.

5.4.1 Évolutions de la chaîne de traitement

Nos différents algorithmes, bien que fonctionnels, pourraient être consolidés, car ils n'ont pas une précision suffisante pour être déployables en production (il faudrait dans l'idéal une précision à 95%). Il y a cependant différentes techniques pour améliorer la précision de nos algorithmes. La technologie principale pouvant être utilisée serait de la détection par apprentissage profond des panneaux. En effet, la détection actuelle des panneaux se fait déjà par apprentissage machine, et en récupérant exactement les panneaux dans les images (liste de pixels qui sont considérés comme étant le panneau par exemple), nos résultats seraient certainement meilleurs. La plupart des formes mauvaises détectées par nos algorithmes seraient évitées de cette manière. En liant cela avec les propriétés géométriques des panneaux selon le dictionnaire créé (voir figure 3.2), on pourrait facilement calculer le centre, la hauteur et l'orientation de ces panneaux.

Nous avions mentionné avec les différents acteurs du projet la possibilité de détecter et classifier les panonceaux de la même manière que les panneaux. Si cela était le cas, il serait intéressant de lier les panonceaux avec les panneaux déjà détectés / à détecter. Cela impliquerait certainement de modifier le dictionnaire créé pour inclure les codes avec les différents panonceaux, ainsi que de se mettre d'accord sur une structure de base de données incluant les panonceaux (actuellement, il n'y a pas de champ dans la base de données pour les panonceaux).

5.4.2 Autres évolutions

Notre travail pourrait s'inscrire dans une certaine mesure dans une démarche d'automatisation dès qu'un utilisateur rajoutera des séquences de photos sur Panoramax. En effet, lorsqu'une séquence est ajoutée sur Panoramax, elle passe par une API de floutage (API *Blur*) pour les visages et plaques d'immatriculation. C'est cette API qui permet de détecter aujourd'hui les panneaux et d'obtenir les imagettes que nous utilisons. La chaîne de traitement développée pourrait donc être incluse dans le processus d'*upload* des séquences sur Panoramax. Une fois la séquence passée par l'API *Blur*, il serait possible de détecter les panneaux via notre chaîne de traitement, les images et imagettes en entrées seraient donc celles de la séquence *uploadée*. Cela permettrait une automatisation de ce processus.

Une autre demande initiale concernait la mise en place d'un web service permettant d'accéder à une base de données en ligne des panneaux détectés. Cette demande a été mise de côté au vu de l'importance et de la complexité de mettre en place la BD PANNEAU avant même de faire le web service associé. De ce fait, une évolution possible serait de développer un web service permettant de requêter la BD PANNEAU.

Conclusion

Le travail accompli a permis de concevoir et d'implémenter une chaîne de traitement complète et robuste, dont les performances ont été rigoureusement évaluées. Le *workflow* débute par la prise en charge des imagettes représentant les panneaux détectés dans les images d'origine. À partir de celles-ci, des informations sont extraites, liées à leurs images d'origine, puis utilisées pour calculer de nouveaux indicateurs. En utilisant toutes ces données, la chaîne de traitement identifie les panneaux identiques, permettant ainsi leur géolocalisation précise et leur dédoublonnage.

Bien que cette chaîne de traitement puisse être encore améliorée, comme le souligne ce rapport, elle témoigne d'une réflexion approfondie et de sa mise en œuvre fructueuse, permettant ainsi d'ébaucher une base de données des panneaux à un niveau significatif. Cela soulève des questions sur la façon d'améliorer la constitution de cette base de données, ainsi que sur ses diverses applications, allant du calcul d'itinéraires à la gestion municipale par les collectivités territoriales et les forces de l'ordre.

En analysant les résultats de ce projet sous un autre point de vue, il est aussi important de souligner la diversité des usages qui pourraient et pourront être faits des photographies *open source* de Panoramax.