

Rapport de recherche
Mission IGN-France
Projet Mapmuxing (Sep-Novembre 2015)

Dieudonné Tsatcha
Ingénieur d'étude et de recherche
Groupe **COGIT** (Cartographie et Géomatique)

1 Introduction du projet

Contexte

Au vu du fait que le tuilage vectoriel est beaucoup utilisé pour la reconnaissance des objets d'une carte, il existe de nos jours un vaste champ de recherche sur des données vectorielles afin de :

- Permettre une recherche facile des différentes informations contenues sur la carte
- Apporter rapidement de nouvelles connaissances aux utilisateurs ;
- Faciliter l'utilisation des approches de « geoprocessing » pour le calcul de la route, la surface des parcelles ;
- Permettre de la personnalisation des contenus de la carte suivant les besoins de l'utilisateur ;
- Permettre à la carte d'aujourd'hui de pouvoir renseigner non seulement des informations spatiales mais aussi temporelles et sémantiques.

Objectifs

Dans l'intérêt de prendre en compte la grande diversité d'informations géographiques disponibles sur le web, il est alors nécessaire de produire un modèle générique et une stratégie de généralisation qui soit adaptée à la grande masse d'objets géographiques disponible sur le web [Gaf12]. La généralisation a été identifiée comme un des fils conducteurs pouvant permettre la mise en place du «web mapping». Pour atteindre ce but, les objectifs suivants sont envisagés :

- la définition d'un modèle de représentation comportant plusieurs représentations de la carte ;
- la définition d'une méthode de transmission adéquate ;
- la proposition des techniques de visualisation des différents niveaux de détails ;
- le développement des algorithmes de transmission incrémentale à l'intérieur d'un format topologique cohérent ceci dans l'initiative de faciliter la reconstruction des niveaux de détails intermédiaires ;
- l'amélioration des interactions entre les utilisateurs et producteurs des données.

Plan : différentes applications à intégrer au projet map-muxing

- Intégration et généralisation des données multi-échelles
- Gestion optimisée des données vectorielles (Dieudonné Tsatcha)
- Optimisation des placements des toponymes (Thibaud Chassin)

2 Gestion optimisée des données vectorielles

L'architecture que nous proposons afin de répondre ce problème de gestion optimisée des données vectorielles est renseignée par la figure 1. Cette architecture se compose de 3 modules essentiels que sont : le stockage, la transmission et le client.

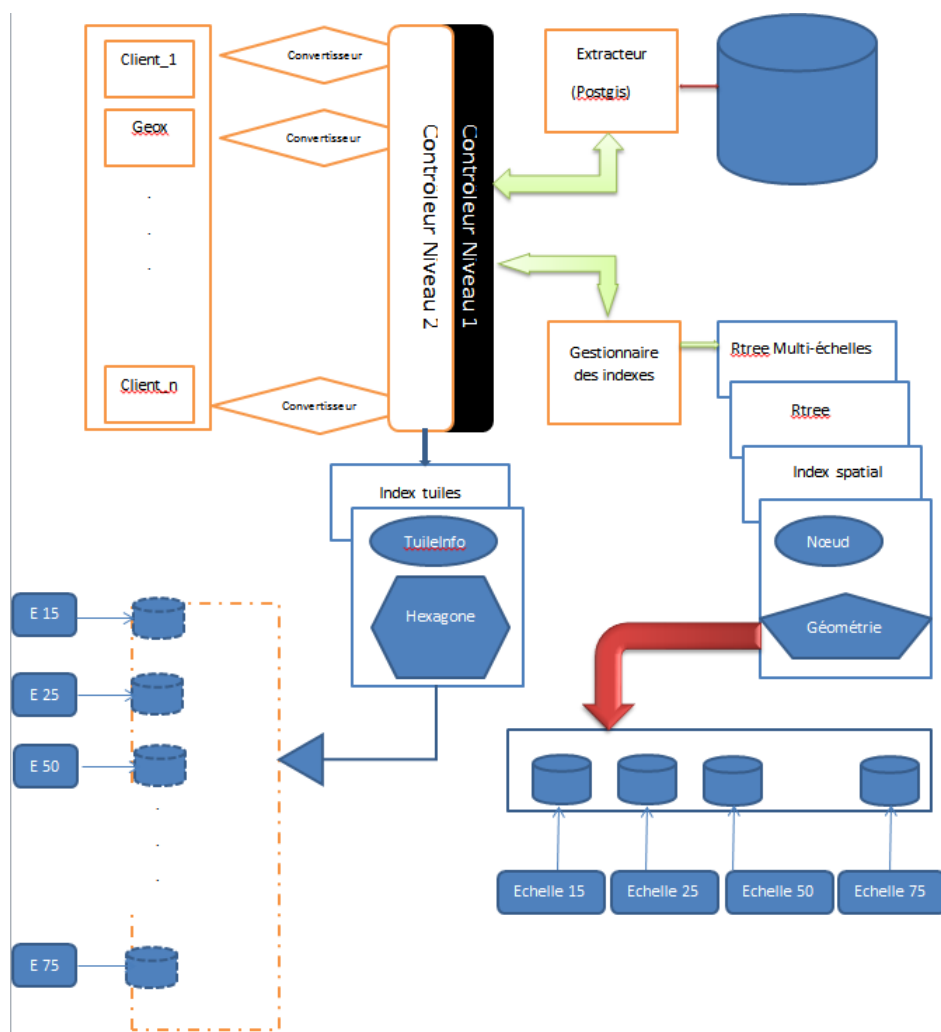


FIGURE 1 – Illustration de l'architecture proposée

2.1 Stockage

Le module de stockage doit mettre en place les approches de calcul dans l'initiative de construire une séquence de représentation de la carte sur des différents niveaux de détails comme abordé dans les techniques de transmission qui le font en une seule fois [Ber01].

Optimisation du modèle de stockage

Il existe plusieurs approches qui permettent de représenter et d'indexer les grandes masses de données géographiques. Les approches les plus répandues sont : les Quadtree, les Rtree, les Btree, etc. L'indexation qui doit implémenter

des données géographiques doit être optimisée afin de réduire la complexité du temps de calcul et de l'espace de stockage.

Parmi toutes les techniques d'indexation proposées, le Rtree paraît la plus optimisée pour des données géographiques que les techniques fondées sur le Quadtree et le Btree.

Le principal défaut des techniques fondées sur du Btree est qu'elles ne sont pas dédiées à la représentation des données multidimensionnelles qui correspondent pourtant aux structururations associées des données géographiques. Également dans ce contexte, les principales raisons qui rendent le Rtree plus optimal que le Quadtree est sa stratégie de mieux optimiser la surface de recouvrement de l'entité spatiale et de permettre l'intégration de la mémoire secondaire. Lorsque nous parlons de la mémoire secondaire, nous faisons allusion à la pagination de la mémoire secondaire (utilisation des systèmes de fichiers de la machine qui l'implémentent).

Les figures suivantes présentent les différents cas qui montrent l'efficacité d'une indexation Rtree par rapport à un QuadTree. En observant correctement les figures 2, on déduit immédiatement que les indexations Rtree gagnent beaucoup d'espaces sur le polygone de recouvrement des entités spatiales. Le polygone de recouvrement utilisé par le Quadtree est un carrée et celui utilisé par le Rtree est un rectangle. En utilisant ces résultats qui montrent l'efficacité d'un Rtree sur QuadTree, nous avons décidé de l'utiliser dans le cadre de cette recherche.

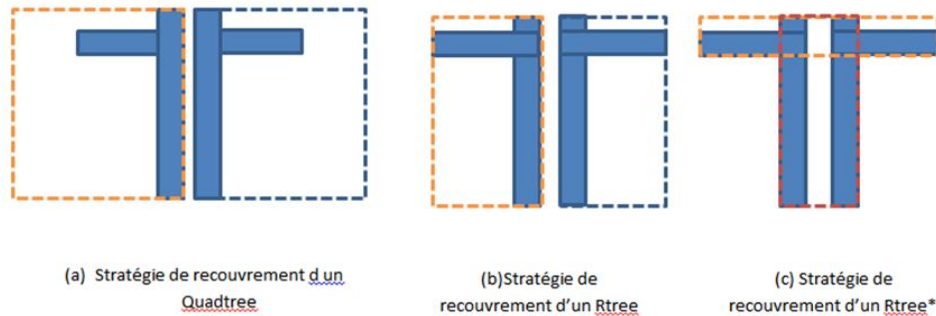


FIGURE 2 – Comparaison du polygone de recouvrement de l'entité spatiale entre un Rtree et un Quadtree

2.2 Indexation fondée d'un Rtree

Une indexation Rtree est basée sur une structure Btree qui stocke plusieurs rectangles de dimensions différentes. Il permet ainsi à l'architecture des Btree de représenter les données multidimensionnelles.

L'indexation Rtree ée initialement proposée par (Guttman, 1984) est une variante des techniques d'indexation conçue pour organiser les collections d'objets spatiaux. Elle représente la donnée spatiale par le plus petit rectangle à deux dimensions qui la recouvre. Sa technique de construction est fondée sur un recouvrement progressif des données spatiales qui produit à la fin un B-arbre

dans lequel chaque nœud de l'arbre correspond au plus petit rectangle contenant d-rectangles (d est le nombre de fils du nœud). Chaque nœud (excepté les nœuds feuilles) stocke un couple d'informations (R,P) où R est le plus petit rectangle et P un pointeur sur les nœuds fils. Les nœuds feuilles ou nœuds terminaux, contiennent des pointeurs qui leur permettent de référencer les données réelles dans la base de données. Chaque nœud feuille se compose également de deux informations (R,O), où R est le plus petit rectangle qui contient l'objet spatial O.

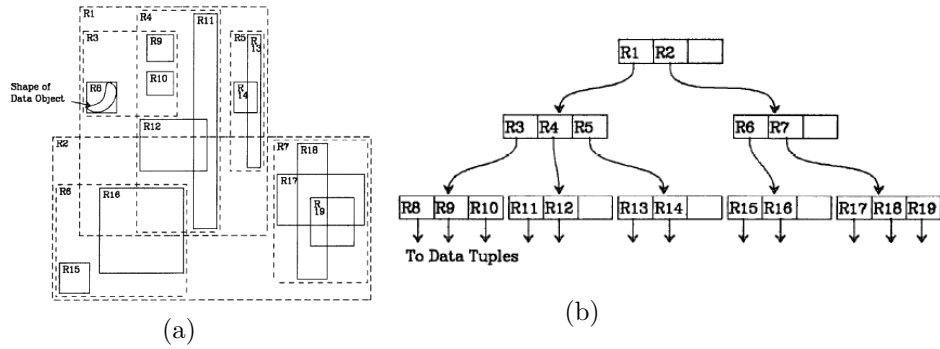


FIGURE 3 – (a) principe de découpage d'un Rtree ; (b) arbre d'indexation d'un Rtree

Dans la structure de données R-tree, les frontières des rectangles formés par des nœuds différents peuvent se chevaucher. Cependant un objet est associé à un seul nœud. La figure 3 (a) est une illustration de la représentation d'un ensemble d'objets spatiaux construits par un R-tree avec son B-arbre (voir figure 3(b)). Les données spatiales réelles sont représentées par des rectangles en traits continus et les nœuds terminaux par des rectangles en trait mixte. La structure de données R-tree respecte un ensemble de propriétés qui permettent d'éviter un dés-équilibre de l'arbre lors des opérations de suppression, d'insertion et de mise à jour de l'arbre. Elle garantit une recherche d'informations dans l'arbre en ordre logarithmique. Les principales propriétés d'un Rtree sont : Soient M et m respectivement le nombre maximum et minimum de fils que peuvent avoir un nœud (excepté les nœuds feuilles), on a toujours $m \geq M/2$.

- Tout nœud (excepté les nœuds feuilles) a au moins deux nœuds fils.
- Toutes les feuilles sont à la même profondeur dans l'arbre.
- L'ordre de recherche ou la taille dans un Rtree est $|\log_m(N)| - 1$ ou N est le nombre de nœuds de l'arbre.
- En fin d'éviter le déséquilibre de ses branches de taille m comprises entre $M/2$ et M où M est le nombre de nœuds fils d'un nœud donné.

2.3 Extension d'un Rtree vers un Rtree multi-échelle

L'indexation Rtree multi-échelle, est l'architecture proposée dans le cadre de ce travail. Elle est fondée sur un Rtree. Dans son principe, chaque échelle de la carte est indexée par un Rtree qui permet de référencer ou d'indexer les contenus de la base de données et afin d'optimiser la transmission vers le client. Par

ailleurs la représentation multi-échelle uniquement ne suffit pas pour répondre au besoin de l'utilisateur qui souhaite visualiser, naviguer, rechercher des informations en temps réel. En ce sens que l'on doit anticiper le comportement du client dans l'intérêt de réduire de plus en plus le temps de transmission et d'affichage des données spatiales au niveau du client. L'un des problèmes majeurs dans les techniques de transmission est la limite des ressources disponibles pour la transmission et la visualisation des données spatiales. Pour apporter des solutions à cette limite des ressources disponibles, plusieurs champs de recherches ont été initiés dans les thématiques suivantes sont explorées :

- Le niveau de détails, c'est une expression empruntée au domaine de la réalité virtuelle. Il permet de représenter l'entité spatiale en différent niveau de complexité comme un maillage progressif. Dans ce travail, il sera principalement dédié aux techniques de choix des échelles à transférer au client.
- Les régions d'intérêt sont utilisées pour contraindre le temps consommé par des différentes opérations effectuées par le client. Concernant, le choix des régions d'intérêts, notre stratégie consistera à mettre à jour régulièrement les données au niveau des caches afin que le client ne soit jamais en défaut de données ou bien en congestion (trop d'information peu pertinent pour le client). Pour se faire, il est important de découper l'espace du client en grille afin de permettre une recherche et une transmission efficaces des données au niveau du cache associé au client. Les données déposées au niveau du cache seront gérées par un contrôleur secondaire afin d'éviter tout risque de congestion au de défaut de cache.
- La transmission progressive, elle consiste à développer des stratégies qui permettent de réaliser la communication efficace entre la base de données ou zone de stockage et des clients. La communication devra ainsi se faire sans interruption ou défaut de données sur tout au niveau du client.

3 Transmission progressive

3.1 Théorie sur la transmission progressive

Plusieurs théories ont été précédemment développées dans la transmission progressive surtout dans le domaine de l'image, c'est le cas des travaux de [Liu09]. Le principal échec énoncé par ces travaux précédents est qu'ils permettent de transmettre uniquement des séries de points les uns après les autres de façon continue avant la transmission définitive [Gaf12]. Cette stratégie ne permet pas d'utiliser ces résultats pour réaliser la transmission des données vectorielles géographiques. En ce sens que, les données géographiques sont énormes et cela peut fragiliser la performance des systèmes.

Plus tard, certains travaux ont permis d'intégrer des grilles aux architectures de transmission, ce qui a abouti à la transmission incrémentale et progressive. Ces transmissions ont été appliquées avec beaucoup de succès sur des images raster vers les clients web comme des prototypes. Ces derniers utilisaient du maillage triangulaire pour la transmission des données vectorielles.

Afin de mieux élaborer une méthode de transmission progressive efficace, il est important de prendre en compte les stratégies suivantes :

- Une stratégie préliminaire du modèle est de pré-calculer les différentes

échelles de la base de données et de les déposer dans le serveur afin de les transférer progressivement vers le client [Ber01]. C'est cette stratégie qui a été initialement proposée dans ce cadre de cette recherche.

- La transmission progressive devra se fonder sur l'architecture des systèmes distribués afin de calculer et de stocker les différentes représentations d'une carte.
- Le serveur devra être à mesure de construire une séquence des représentations de la carte à différents niveaux de détails car une transmission progressive est développée pour envoyer les différents niveaux de détails au client en un seul temps.
- Le client de son côté doit avoir un mécanisme de reconstruction des données, ceci dans l'initiative d'avoir les différentes couches graphiques pour ainsi afficher la représentation complète à chaque étape (correspondant à la transmission d'un niveau de détail donné). Un algorithme d'intégration devra être développé et implémenté afin de reconstruire le «dataset» correspondant de la représentation sollicitée.

De nos jours plusieurs axes de recherche sont explorés pour améliorer les techniques d'indexation donc les axes les plus porteurs sont les suivants :

- la définition d'un modèle consistant des multiples représentations de la carte ;
- la définition de la technique de transmission ;
- le développement d'une interface graphique et un d'outil de visualisation dynamique afin d'afficher les différentes couches correspondant à chaque niveau de détail.

4 Proposition d'une transmission progressive

Dans le modèle de l'architecture proposée (voir figure 1) deux contrôleurs ont été implémentés pour réaliser cette communication. Le contrôleur de niveau I formé de deux composants essentiels, le gestionnaire des indexes et l'extracteur des données dans la base de données. Ce dernier recherche initialement des données géographiques dans l'indexation multi-Rtree en utilisant le gestionnaire des indexes et si ces données ne sont pas présentes, il utilise son extracteur pour les récupérer de la base de données Postgresql et les chargées dans l'indexation multi-Rtree. Dans le cas que ces données sont présentes dans l'indexation multi-Rtree celles-ci sont directement envoyées au contrôleur de niveau II.

4.1 Gestionnaire des indexes

Il est chargé de construire et de gérer les indexes construits dans la modélisation multi-échelle. Cela est rendu possible par la classe ScaleInfo. Cette classe permet de générer l'échelle associée à une table de la base de données ou bien à une valeur quelconque fournie en entrée. Deux informations essentielles permettent de représenter une échelle. Nous avons la valeur minimale qui correspond à l'échelle stockée dans l'index et la valeur maximale qui correspond à la valeur au-dessus de laquelle on change d'échelle. Les différents intervalles des échelles réalisées sont les suivantes : $([15, 25], [25, 50], [50, 75], [75, 100], [250, 1000], [1000, \infty])$. Pour une valeur d'échelle qui n'a pas été renseignée on prendra la valeur d'échelle égale à 15. Par ailleurs, si la valeur d'échelle est supérieure à 1000, on prendra

la valeur d'échelle égale à 1000. Les fonctions que le gestionnaire d'index implémente sont les suivantes :

- Création d'index
Cette fonction consiste à fabriquer un répertoire dans lequel est stocké toutes les informations d'indexation. Le répertoire ainsi fabriqué répond aux problématiques de bases de données sécurisées et multi-connexion. Cette technique s'appuie sur des stratégies proposées par le projet « sleep-cat »
- Suppression d'index
Elle consiste à supprimer complètement le répertoire qui contient cet index.
- Utilisation d'un index
Cette fonction permet de demander un index au gestionnaire afin de pouvoir l'utiliser efficacement. C'est à dire de demander une échelle quelconque et de pouvoir charger un enregistrement, supprimer un enregistrement, mettre à jour un enregistrement, rechercher un enregistrement. Un enregistrement dans notre modèle est un couple (Nœud, Géométrie). Le nœud permet de référencer de façon unique une géométrie dans l'index. La recherche d'une information se fait à partir du champ nœud. Ce dernier doit être représenté de telle sorte tous les éléments dans la base d'index ont des nœuds différents. La mise en œuvre efficace de ces techniques sur l'utilisation des indexes s'appuient sur un fichier « scaleInfo » contenu dans le dossier « spatial » des sources du projet. Dans ce fichier est enregistré toutes les informations associées à la bonne gestion de l'index.

4.2 Extracteur des données dans la base de données

Le stockage de ces données utilisées dans le cadre de ce projet est rendu possible par la base de données PostgreSQL qui stocke des données relationnelles. Les contenus sont extraits via l'extracteur en utilisant les techniques de connexion persistante associée au graveur. Dans notre architecture proposée, il sera possible d'intégrer plusieurs autres bases de données qui seront toutes référencées par un même extracteur en indiquant uniquement les informations de connexion de la base de données sollicitée.

4.3 Contrôleur Niveau II

Il s'assure une communication fluide entre le contrôleur de niveau I et le client. Son rôle permet d'éviter d'aller rechercher de la donnée à la demande du client. Il est alors important dans les architectures de transmission de toujours mettre à la disposition du client, un flot de données précédemment pré-calculé. Cependant, ce flot de données peut avoir deux impacts majeurs :

- Le premier impact est qu'il peut créer une congestion de données suite à un flot de données important. Dans ce cas, il faut fournir uniquement la quantité de données qui peut intéresser directement le client. Les données peu importantes sont directement retirées de la base d'affichage du client afin d'avoir toujours un client moins lourd.
- Le deuxième impact est le défaut de données. Le client ne devra jamais attendre qu'on recherche ou calcule de la donnée avant afin qu'il s'approvisionne. Le contrôleur de niveau II doit comprendre et analyser les

intentions (zoom, pan) du client afin de pouvoir lui fournir les flots données à la demande.

4.4 Gestion du flot d'information entre le client-contrôleur Niveau II

Enfin de mieux gérer ce flot de données entre le client et le contrôleur du niveau II. Il est impératif de déterminer une stratégie de tuilage efficace pour cette transmission sans toutefois perturber les utilisateurs de la réalité. De toutes les géométries régulières utilisés dans le tuilage, l'hexagone a été montré être le plus compact et pouvant représenter les données du globe terrestres [Kev11]. Les preuves de cette compacité nommée « quantization » s'appuie sur des erreurs d'encodage des nombres utilisés pour représenter la cellule d'une grille. Ces erreurs peuvent se propager tout au long de la construction d'une grille et devenir ennuyeux au modèle. Déplus, comme ce projet vise à inclure des cartes numériques du globe dans lequel on souhaiterait observer qu'une partie de ce globe tout en interagissant, en visualisant, en manipulant et en modélisant les gros volumes des données spatiales. Il s'avère alors important de s'appuyer sur des techniques d'indexation performantes. Le tuilage hexagonal dans cette vision paraît intéressant. Dans la littérature il existe 3 types de grille hexagonale appliqués intégrant la multi-résolution. Nous avons des grilles en couverture [Kev11]. L'aperture 3 exige un facteur d'échelle de précision de $(1 + \sqrt{3})$ et une rotation de 30° (voir figure 4(a)), l'aperture 4 exige un facteur d'échelle de précision de $(1 + \sqrt{4})$ et sans rotation (voir figure 4(b)), enfin l'aperture 7 exige un facteur d'échelle de précision de $(1 + \sqrt{7})$ et une rotation de $19,1^\circ$ soit $(\arcsin \sqrt{\frac{3}{28}})$ (voir figure 4(c)).

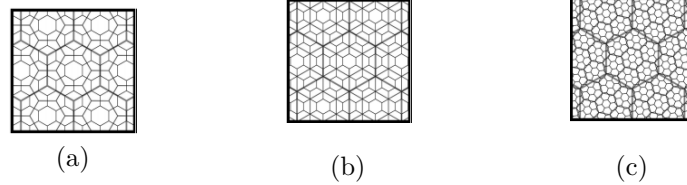


FIGURE 4 – (a) couverture 3 (b) couverture 4 et (c) couverture 7

Dans le cadre de ce travail, le tuilage utilisé est l'aperture 4 car elle paraît plus compacte pour la représentation des données du globe. Par contre, nous n'avons pas abordé dans nos travaux actuels la gestion du tuilage multi-résolution sur l'aperture 4.

Compte-tenu du fait que le contrôleur de niveau II est chargé de contrôler les flots de données qu'a besoin le client pour un affichage correct. Nous avons identifié sur chaque échelle la zone correspondante à un affichage correct du client. Cette zone se détermine à partir du point d'intérêt identifié à l'écran du client. Ce point d'intérêt correspond à la position du curseur sur l'écran et la zone d'affichage correspond à l'écran du client.

Ces informations (le point d'intérêt et la zone d'affichage) nous permettent de fabriquer un cadrage de données. Un cadrage de données vectorielles est un

ensemble de 9 cadres dont un seul cadre est choisi à tout moment comme cadre central ou zone d’affichage. Lorsque le client désire des données nécessaires à son écran, un cadrage lui est fourni et parmi les 9 cadres un seul est destiné à l’affichage et le 8 autres sont transmis dans le cache. Les données de ce cadrage sont ainsi tuilées par une technique d’indexation que nous avons proposée dans ce travail. Cette technique d’indexation est mieux détaillée à la section 5.2.

4.5 Principes de fonctionnement de la transmission progressive proposée

A l’initialisation du système, le contrôleur II met à la disposition du client toutes les données de l’échelle maximale nécessaire pour recouvrir 9 fois l’écran du système (appelé zone de cadrage). Ce contrôleur tuile également ce cadrage pour toutes échelles déterminées dans le système. Pour déterminer les données, le contrôleur du niveau II se renseignera à la base d’indexation Hexatree. Ce dernier retournera les données du cadre central si ce cadre est tuilé dans l’index, si ce n’est pas le cas le contrôleur du niveau II demandera au contrôleur de niveau I de lui fournir non seulement les données associées à ce cadre mais tout le cadrage dans lequel ce cadre est le cadre central.

– Gestion d’un zoom

La gestion d’un zoom revient à rechercher dans l’indexation HexaTree, les données demandées pour l’échelle choisie. Pour se faire, le client se renseigne chez le contrôleur du niveau II.

– Translation

En considérant le sens de déplacement du curseur, il est possible de détecter s’il est question d’une : translation verticale, oblique, et horizontale. La nature de la translation permet d’identifier lequel des cadres de la grille qui doit être sollicité ou ajouté chez le client. La translation, modifie généralement le point d’intérêt. Pour prendre en compte la translation dans notre modèle, nous procédons comme suit :

1. Déterminer le point d’intérêt.
2. Identifier parmi les cadres contenus dans l’indexation Hexatree lequel contient ce point
3. Si aucun cadre ne contient ce point d’intérêt on a un défaut de cache. Il est important que le contrôleur du niveau II reconstruise le cadrage associé à ce point et demande au contrôleur du niveau I de lui fournir toutes les données associées à ce cadrage pour toutes les échelles. Pendant la transmission du cadrage, le cadre centra est directement transmis au client.
4. Si le cadre est présent dans la l’indexation Hexatree, le contrôleur du niveau II retourne ce cadre au client et met à jour les données d’indexation Hexatree. Il s’appuie sur des techniques de gestion optimisée de la navigation présentée dans la section 5.1.

5 Clients

Deux principaux facteurs doivent être respectés par le fournisseur de données au client. Il devra éviter que le client soit en congestion de données ou à défaut de données. Ainsi, il faut lui fournir les données en tenant compte de la taille des données qui soit suffisante au client à tout moment. Pour cela on doit avoir les caractéristiques du client. Les caractéristiques du client au niveau visualisation doivent sa hauteur et sa largeur. Ces deux paramètres permettent de déterminer le volume de données qui sont suffisantes au client. Pour permettre un fonctionnement efficace de données au client, le producteur doit mettre à l'avance les données qui sont nécessaires au client pour son bon fonctionnement. L'architecture proposée précédemment décrit correctement la communication entre le client et son producteur des données. Dans le cadre de ce travail, le producteur de données est le serveur de base de données.

5.1 Techniques de gestion optimisée de la navigation de l'architecture proposée

Gestion de la phase de chargement

Initialement les données renseignées au niveau des tables de la base de données sont complètement tuilées au côté du serveur. L'indexation utilisée pour ce tuilage est Rtree. Le choix du Rtree a été présenté dans la section 2.2. Ceci est dans l'intérêt d'éviter plusieurs transferts entre le système et la base de données et aussi de mieux organiser les différents contenus qu'on souhaite transmettre au client. Dans notre architecture nous avons proposé une extension du Rtree nommée Rtree multi-échelles qui permet d'organiser les données en plusieurs échelles afin de faciliter le choix de données en fonction des échelles.

Gestion de la première connexion

A la première connexion, les paramètres de visualisation du client sont récupérés afin de construire un cadrage. Un cadrage est un ensemble de 9 cadres de forme rectangulaire dont le cadre central est identique à l'écran du client. Le cadre central correspondant à l'écran du client est le cadre 4 dans la figure ci-dessous.

Ce cadrage permet de déterminer parmi les éléments tuilés dans le Rtree ceux qui seront transmis dans l'indexation Hexatree décrit plus loin. La stratégie est non bloquante, au même moment que se réalise le tuilage par une indexation Hexatree les éléments nécessaires par le cadre 4 ou affichage correct du client lui sont directement transmis. Par ailleurs, les huit autres cadres sont utilisées comme des caches afin de faciliter la recherche des données suite à une quelconque opération (zoom ou pan) effectuée au niveau client.

5.2 Indexation Hexatree

Une indexation Hexatree se fonde sur une technique de découpage récursif des entités spatiales par une géométrie régulière. Tenant compte du fait que

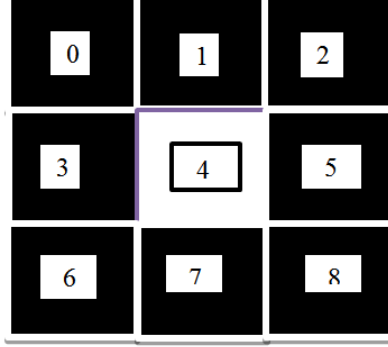


FIGURE 5 – Illustration d'un cadrage

les techniques de découpage peuvent produire pertes d'informations. Il était important pour nous de choisir la géométrie régulière qui paraît plus dense dans la reconstruction des entités spatiales.

D'après la littérature et plus précisément ceux des travaux de [Kev11], l'hexagone paraît être la structure géométrique la plus optimale pour le découpage des objets de globe terrestre. Quand nous parlons de l'hexagone nous faisons aussi allusion à toutes ces stratégies multi-hiérarchiques qu'il permet de réaliser. Dans le cadre de ce travail, la technique d'indexation multi-échelles envisagée est l'aperture ou couverture 4.

Dans la modélisation d'un Hexatree, les paramètres de visualisation du client sont importants car ils permettent de déterminer le rapport de projection entre les paramètres de visualisation avec la plus petite échelle disponible dans le processus de transmission. Ce rapport de transmission permet de calculer les paramètres de visualisation des autres échelles disponibles dans le modèle de transmission. Ainsi, on doit définir l'enveloppe convexe associée à chaque échelle ou laisser le système les déterminer automatique lors de l'indexation Rtree multi-échelle. Cependant, les paramètres de visualisation de l'écran sont fournis par le client. Ces paramètres permettent de déterminer également son enveloppe convexe qui sera utilisé pour déterminer le rapport de transmission. La figure illustre, le principe de détermination du rapport de transmission. Ce rapport doit être constant afin de ne pas troubler la perception du client. $AB/ab = EF/ef$ également, $BC/bc = FG/fg$, se rapport de transmission doit être conservé afin de ne perturber les transferts des données à l'endroit de l'utilisateur.

A partir de cette stratégie de transmission sur rapport constant, on peut réorganiser le tuilage au niveau du client en fonction de la résolution de son écran. Puisque le cadre de visualisation se compose de 4 hexagones correctement construits de façon à recouvrir complètement le cadre. Il est alors possible déterminer le rayon de l'hexagone. Ces hexagones contiennent les composants des entités spatiales découpées pour la transmission. La figure ci-dessous présente la stratégie de construction des hexagones qui recouvrent un cadre ou la fenêtre de visualisation du client.

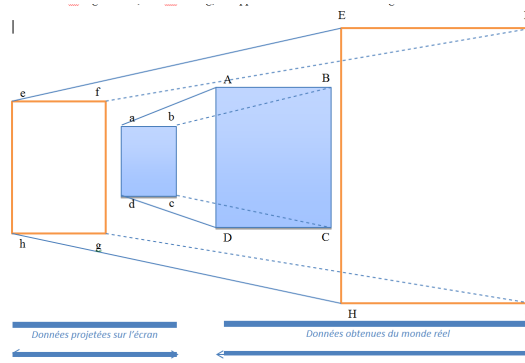


FIGURE 6 – Stratégie de détermination du rapport de transmission

Présentation de la stratégie de transmission

La fenêtre du client est décomposée en hexagones illustrée de la façon suivante. Ces derniers sont alors projetés dans les données d'une échelle donnée en considérant le rapport de transmission.

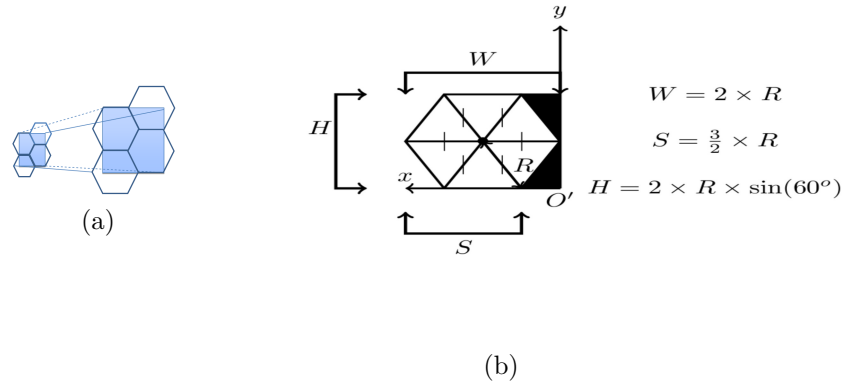


FIGURE 7 – (a) découpage de l'écran du client ; (b) les paramètres de l'hexagone

Soient l , L respectivement la longueur et la largeur de l'écran. Les paramètres de l'hexagone 8(b) sont calculés comme suit : $H = L * 2/3$; $R = l/2$; $S = 3 * R/2$; $W = R * 2$;

Insertion d'une géométrie dans un HexaTree

Une fois le cadrage projeté dans une échelle donnée, on identifie la partie des entités spatiales qui sont contenue dans le cadrage. Cette partie de l'entité spatiale est alors répartie à l'intérieur des 9 cadres que composent le cadrage. Chaque cadre, récupère ainsi la partie de l'entité spatiale qu'il contient. Le contenu de l'entité spatial délimité par chaque cadre est directement reparti

parmi les 4 hexagones que composent dans le cadre (voir figure 8(a)). La figure (8(b)) illustre le principe de découpage des entités spatiales contenue à une échelle quelconque.

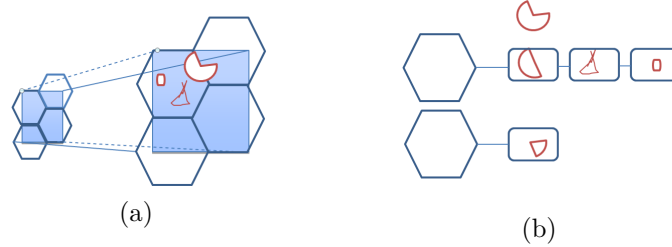


FIGURE 8 – projection des hexagones sur des échelles

La plus grande qualité de cette indexation est de permettre une possibilité de pagination de la mémoire en indexant le système de fichiers des machines qui les implémentent. Un index est généralement formé de deux parties essentielles à savoir ; la clé et la valeur. La clé permet de rechercher une valeur qui lui est associée en $\oplus(1)$. Par ailleurs un index spatial est composé de deux composants : le nœud et la géométrie. Ces deux composantes sont sauvegardées dans des fichiers de la machines et gérées par des techniques associées aux bases de données. Ces techniques permettent de définir plusieurs connexions par conséquent permettre la gestion de l'accès concurrent de la mémoire et aussi sécurisé les contenus. Dans le cas de ce travail nous avons utilisé les techniques de base de données proposés par le projet Sleepcat. L'arbre Hexatree permet de fabriquer un nœud associé à une géométrie qui permet d'organiser les contenus dans le système de fichiers et de pouvoir reconstruire une entité géométrique à la fin de la transmission. Tout au long de la transmission les informations associées au nœud sont régulièrement modifiée.

Dans notre modèle, un nœud est formé de trois composantes : les informations de base de données, les informations sur des échelles et les informations obtenues dans l'Hexatree. Dans le système de fichiers de la machine les répertoires associés à chaque échelle sont créés et chaque répertoire comprend 9 fichiers pour indexer les 9 cadres d'un cadrage donné. Les attributs des différents composants sont détaillés comme suit :

Les attributs récupérés de la base de données

- le nom de la table
- le srid
- son numéro dans la table
- sa signature (géométrie ou empreinte spatiale).

Les attributs provenant de l'échelle

- Le nom de l'échelle qui est composé du mot scale + valeur
- L'identifiant de l'échelle qui permet d'identifier l'échelle (15, 25, 50, 75, 100, 1000) dans le cas qu'on a une échelle inférieure à 15, l'identifiant considéré est 15

et par contre si l'échelle est supérieure à 1000 alors l'identifiant considéré 1000.

- Un même identifiant (id1) est considéré pour des échelles comprises entre $[id1, id2[$ ou id1, et id2 appartiennent à (15, 25, 50, 75, 100, 1000) et id2 est le premier identifiant supérieur à id1.

Les attributs de l'Hexatree

L'Hexatree permet de renseigner les informations suivantes au nœud lors de la transmission :

- l'indice en abscisse (indiceX) de l'hexagone contenant le fragment de l'entité spatiale
- l'indice en ordonné (indiceY) de l'hexagone contenant le fragment de l'entité spatiale
- La génération de l'hexagone qui permet de proposer la transmission multi-hiérarchique des fragments des cellules composant l'entité spatiale.

Gestion de la navigation opérée par un utilisateur

Dans l'objectif d'éviter de tout mettre à jour ou de recréer l'Hexatree à chaque opération effectuée par le client. Il est important d'anticiper le chargement des données nécessaire pour l'affichage correct du client. Pour se faire, nous avons proposé une stratégie de gestion de chevauchement des cadrages. Le chevauchement des cadrages permet d'identifier à tout moment la liste des cadres qu'il faut déterminer en fonction de ceux qui ont déjà été tuilés dans l'indexation Hexatree.

Soient les cadrages rouges et noirs de la figure (9) 8, où le cadrage noir représente le cadrage des contenus tuilés dans l'Hexatree et le cadrage rouge ceux déterminés à la suite d'un nouveau changement du point d'intérêt du client. Un cadrage est représenté par un vecteur de transfert T (voir figure 9, où la valeur $T(i, j)$ peut désigner que le cadre est déjà chargé (dans ce cas $T(i, j)$ à une valeur différente de -1) et sinon $T(i, j)$ prend une la valeur égale à -1.

Le vecteur 1 désigne, le vecteur de transfert du cadrage rouge avant le croisement avec celui du cadrage contenu dans l'Hexatree. Ainsi, l'algorithme proposé permet de prendre en entrée un ensemble d'indices du nouveau cadrage initialisée à -1 pour chaque entrée. Lorsqu'un cadre du nouveau cadrage (rouge) correspond exactement à un cadre de l'ancien cadrage précédemment tuilé, on remplace la valeur -1 par le numéro du cadre de cet ancien cadrage.

En considérant, l'exemple illustré par la figure 9, l'ancien cadrage est en noir et le nouveau cadrage est en rouge. Nous aurons : la valeur de l'indice 0 du nouveau cadre prenant la valeur 7 et celle à l'indice 1 prenant la valeur 8. Le reste des indices ayant la valeur -1 doivent être recherché afin d'être tuilés dans l'Hexatree.

Avant le chevauchement le vecteur de transfert du cadrage rouge est renseignée comme suit :

$$T^{(i,j)} = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix} \quad (1)$$

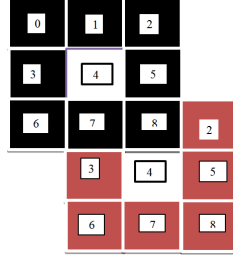


FIGURE 9 – Stratégie de mise à jour

Après le chevauchement le vecteur de transfert du cadrage rouge est modifié comme suit :

$$T^{(i,j)} = \begin{pmatrix} 7 & 8 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix} \quad (2)$$

Les résultat de ce transfert montre qu'il faut aller chercher les contenus des cadres 2, 3, 4, 5, 6, 7 et 8 et que par contre ceux des cadres 0 et 1 sont déjà présents dans l'indexation Hexatree.

6 Conclusion

En définitive, ce travail met en évidence la faisabilité d'une stratégie de tuilage des données vectorielles multi-échelles. Il présente de façon succincte les trois approches utilisées qui sont (1) la détermination du niveau de détails, (2) le choix du point d'intérêt et (3) la stratégie de transmission afin de réaliser le transfert de données entre le serveur et le client. La détermination du niveau de détails se fait en réalisant une structuration multi-échelle des données vectorielles provenant d'une base de données Postgresql à l'aide d'une indexation Rtree. L'intérêt d'intégrer cette indexation est de pouvoir réduire considérablement les temps de transfert et de requêtes consommés sur des bases de données qui n'ont pas une structuration en échelle des données géographiques. Concernant le choix du point d'intérêt, cela nous permet d'identifier la position du curseur sur l'écran du client afin de construire le cadrage des données qui soit nécessaire à tuiler dans l'Hexatree. Ces données tuilées par l'Hexatree sont plus tard transmises au client pour affichage. En ce qui concerne la transmission, elle paraît être la tâche la plus difficile dans la gestion des données vectorielles, car elle exige de contrôler et de comprendre les comportements des clients et ceux des producteurs des données (serveurs).

En ce sens que l'inefficacité de contrôle et l'absence de compréhension adéquate des comportements du client peut compromettre la performance de la transmission en créant deux problèmes majeurs à savoir la congestion de données et le défaut (absence) de données à transmettre au client. Il est donc important de mettre en œuvre des stratégies qui permettent d'éviter ces deux problèmes. L'évitement de congestion de données au client, exige de ne pas trop surcharger le client lors de la transmission et par contre l'évitement de l'absence de données (le défaut de données) suggère de toujours mettre à la disposition du client de la donnée avant qu'il en demande.

La réponse à ces problèmes de congestion et de défaut de données, nous a conduit à construire des cadrages en fonction des points d'intérêts qui permettent de pre-calculer des flots de données qui seront cachés au client.

Bien que les premiers résultat de ces travaux sont assez concluant, d'autres taches sont encore envisagées afin de rendre le modèle beaucoup plus robuste. Les taches que nous avons identifiées sont par exemple, l'intégration des priorités d'affichage sur des différentes couches thématiques en un moment donnée, une transmission directe entre le serveur et client en évitant les structure de données bloquantes (les listes, etc.), intégration des stratégies de communication parallèle entre la processus d'affichage et ceux qui permettent de reconstruite l'Hexatree.

Références

- [Ber01] M. Bertolotto. Progressive transmission of vector map data over the world wide web. *GeoInformatica*, pages 34–373, 2001.
- [Gaf12] J. Gaffuri. Toward web mapping with vector data. In *Geographic Information Science*, volume 7478, pages 87–100, 2012.
- [Kev11] S. Kevin. Discrete global grid systems. archives of photogrammetry, cartography and remote sensing. pages 363–376, 2011.
- [Liu09] H. Liu. Progressive vector data transmission based on overall effects. In *Fourth International Conference on Networked Computing and Advanced Information Management*, pages 604–608, 2009.